

## Hochschule für Technik, Wirtschaft und Kultur Leipzig

# Studienordnung Bachelorstudiengang Informatik

- StudO-INB -

Fassung vom 27.08 2019 auf der Grundlage von §§ 13 Abs. 4, 36 SächsHSFG

Aus Gründen der besseren Lesbarkeit wird auf die gleichzeitige Verwendung männlicher und weiblicher Sprachformen verzichtet. Maskuline Personenbezeichnungen in dieser Ordnung gelten gleichermaßen für Personen weiblichen Geschlechts.

### Inhaltsverzeichnis

§ 1	Geltungsbereich .....	2
§ 2	Studienziel.....	2
§ 3	Zulassungsvoraussetzungen.....	3
§ 4	Aufbau und Inhalt des Studiums.....	3
§ 5	Studienberatung .....	5
§ 6	Schlussbestimmungen .....	6

## **§ 1 Geltungsbereich**

(1) Diese Studienordnung legt auf der Grundlage der zugehörigen Prüfungsordnung das Studienziel, die Zulassungsvoraussetzungen, den Aufbau und den Inhalt des Bachelorstudiengangs Informatik (INB) an der Fakultät Informatik und Medien (IM) der HTWK Leipzig fest.

(2) Der Verlauf des Studiums ist im **Studienablauf- und Prüfungsplan** (vgl. PrüfO-INB **Anlage 1 und 2**) ausgewiesen. Er hat insoweit empfehlenden Charakter, als bei seiner Beachtung der Bachelorgrad innerhalb der Regelstudienzeit von 6 Semestern erreicht werden kann. Der Studienablauf- und der Prüfungsplan wird durch die **Modulbeschreibungen** im Modulhandbuch (vgl. **Anlage 1**) konkretisiert.

(3) Ziel, Zulassung, Aufbau und Inhalt der in das Studium integrierten berufspraktischen Tätigkeit (Praxisphase) regelt die **Praktikumsordnung** (vgl. **Anlage 2**), die Bestandteil dieser Studienordnung ist.

(4) Ein Teilstudium ist mit reduziertem Inhalt auch über einen verkürzten Zeitraum von maximal 2 Semestern möglich.

## **§ 2 Studienziel**

(1) Das Studium soll auf die berufliche Tätigkeit vorbereiten und die erforderlichen fachlichen Kenntnisse, Fähigkeiten und Methoden so vermitteln, dass der Student zu wissenschaftlicher Arbeit, zu selbständigem Denken und zu verantwortungsbewusstem Handeln befähigt wird. Neben der Vermittlung berufsbezogenen Wissens soll das Studium auch die Grundlage für weiterführende wissenschaftliche Studien schaffen.

(2) Dem Studenten soll die Fähigkeit vermittelt werden, wissenschaftliche Methoden und Erkenntnisse selbstständig zur Analyse und Lösung von Problemen auf dem Gebiet der Informatik anzuwenden. Das analytische, logische Denken in Strukturen und Konzepten soll ausgeprägt werden. Dazu erwirbt der Student grundlegende Fachkenntnisse, praxis- und anwendungsbezogene Fähigkeiten auf Gebieten der Praktischen, Technischen, Angewandten und Theoretischen Informatik vor dem Hintergrund der Planung und Realisierung komplexer Systemlösungen. Darüber hinaus werden übergreifende Fach- und Sozialkompetenzen (Schlüsselqualifikationen) und Strategien für lebenslanges Lernen vermittelt.

(3) Im Bachelorstudiengang Informatik liegen die fachlichen Schwerpunkte auf folgenden Gebieten:

- Zusammenspiel von Hardware und Software in modernen Rechnerarchitekturen

- Entwicklung von Software unter Einsatz fundierter Kenntnisse auf den Gebieten Rechnerarchitekturen, Betriebssysteme, Netzwerke und Datenbanken für klassische und mobile Systeme
- Entwicklung von Applikationen und Informationssystemen für Betriebs- und Geschäftsprozesse
- Entwicklung nutzerorientierter Interaktionsoberflächen

Der Bachelorstudiengang Informatik befähigt seine Absolventen zu einer aktiven Gestaltung komplexer medienbezogener informationsverarbeitender Prozesse in allen Bereichen der Gesellschaft. Er eröffnet gut ausgebildeten Fachleuten national und international ausgezeichnete berufliche Entwicklungschancen, und zwar hauptsächlich

- in Unternehmen, die Software/Hardware herstellen und/oder vertreiben,
- bei Software- und Computersystemanwendern (Industrie, Handel, Banken, Versicherungen),
- in Telekommunikationsunternehmen
- in Beratungs- und Dienstleistungsunternehmen,
- in Institutionen zur Aus- und Weiterbildung.

Die Kompetenzprofile der Absolventen des Studienganges werden im Diploma Supplement konkretisiert. Das Muster des Diploma Supplement wird im Internetportal der HTWK Leipzig unter [www.htwk-leipzig.de](http://www.htwk-leipzig.de) veröffentlicht.

(4) Das Studium wird mit dem Erwerb des ersten berufsqualifizierenden Abschlusses „Bachelor of Science“, abgekürzt „B.Sc.“, beendet.

### **§ 3**

#### **Zulassungsvoraussetzungen**

(1) Die Zulassung zum Studium bestimmt sich nach den einschlägigen hochschulrechtlichen Bestimmungen, insbesondere nach dem Sächsischen Hochschulfreiheitsgesetz, dem Sächsischen Hochschulzulassungsgesetz und der Sächsischen Studienplatzvergabeverordnung sowie nach der Immatrikulationsordnung und Auswahlordnung der HTWK Leipzig.

(2) Über die Gleichwertigkeit von nachgewiesener Vorbildung und Hochschulzugangsberechtigung entscheidet im Zweifel der Prüfungsausschuss.

### **§ 4**

#### **Aufbau und Inhalt des Studiums**

(1) Das Studium wird in der Regel zum Wintersemester aufgenommen.

(2) Die Studieninhalte werden in Modulen vermittelt (modularer Aufbau). Module bezeichnen einen Verbund zeitlich begrenzter, in sich geschlossener, inhaltlich oder methodisch ausgerichteter Lehrveranstaltungen. Jedes Modul wird mit einer Modulprüfung abgeschlossen.

sen, die nach Maßgabe des Prüfungsplans aus einer oder mehreren Prüfungen bestehen kann. Für erfolgreich absolvierte Module werden entsprechend ihrem hierzu erforderlichen Zeitaufwand für

- a) die Teilnahme an Lehrveranstaltungen,
- b) die Vor- und Nachbereitung von Lehrveranstaltungen,
- c) die Ableistung der Praxisphase,
- d) das Selbststudium sowie
- e) die Vorbereitung auf und die Ablegung von Prüfungen

(sog. Arbeitslast oder workload) Punkte nach dem **European Credit Transfer and Accumulation System** (ECTS-Punkte, Leistungspunkte) vergeben. Ein ECTS-Punkt entspricht für einen durchschnittlich leistungsfähigen Studenten einer Arbeitslast von 30 Zeitstunden.

(3) Vermittlungsformen in Lehrveranstaltungen können insbesondere Vorlesungen, Übungen, Seminare und Praktika sein. Nach Maßgabe der Modulbeschreibungen können Lehrveranstaltungen auch in einer Fremdsprache abgehalten werden.

(4) Der erfolgreiche Abschluss des Studiums erfordert den Erwerb von 180 ECTS-Punkten. Nach Maßgabe des Studienablauf- und des Prüfungsplans sind dabei aus den Pflichtmodulen 140, aus den Wahlpflichtmodulen 40 ECTS-Punkte zu erbringen. Im Rahmen der fachbezogenen Fremdsprachenausbildung müssen 4 ECTS-Punkte erworben werden.

(5) Die Module werden nach

- a) **Pflichtmodulen**, die jeder Student zu belegen hat und
- b) **Wahlpflichtmodulen**, unter denen der Student innerhalb des Modulangebots des Studiengangs auswählen kann und in bestimmten Umfang auswählen muss, und
- c) **Zusatzmodulen**, die der Student über das Modulangebot des Studiengangs hinaus belegen kann,

unterschieden. Weitere Einzelheiten zu den Modulen ergeben sich aus den Modulbeschreibungen.

(6) Das Studium ist in Grundstudium und Hauptstudium gegliedert - mit jeweils drei Semestern. Ersteres besteht ausschließlich aus Pflichtmodulen, in denen Wissen vermittelt wird, das für das Verständnis nachfolgender Module wesentlich ist. Im Hauptstudium ist der Pflichtanteil relativ gering; es dominieren die Wahlpflichtmodule. Diese sind strukturiert in drei Bausteine, und zwar

- a) Baustein **Technologien für Softwaresysteme**
- b) Baustein **Programmiertechniken**
- c) Baustein **Technische Systeme**

Jeder Baustein besteht aus vier Modulen. Zwei Bausteine sind pflichtgemäß zu absolvieren, wobei ein Baustein als absolviert gilt, wenn mindestens drei der in ihm enthaltenen Module erfolgreich absolviert sind. Die absolvierten Bausteine werden im Zeugnis ausgewiesen.

(7) Die Zulassung zu Wahlpflichtmodulen hat der Student auf dem Wege der Einschreibung spätestens bis zum Ende der Einschreibungsfrist im vorherigen Semester zu beantragen. Über die Zulassung entscheidet das Prüfungsamt im Einvernehmen mit dem Studiendekan unter Berücksichtigung kapazitätsbedingter Möglichkeiten. Im Fall der Wahl eines Moduls an einer anderen Fakultät bzw. Einrichtung erfordert eine Zulassung deren Zustimmung. Stellt der Student keinen Antrag, kann ihn das Prüfungsamt von Amts wegen zulassen. Die Zulassung ist unanfechtbar.

(8) Anzahl und Inhalt der angebotenen Wahlpflichtmodule können verändert werden, wenn die Berücksichtigung des aktuellen wissenschaftlichen Erkenntnisstandes oder eine Verlagerung der Lehr- und Forschungsschwerpunkte dies erfordern. Werden für ein Wahlpflichtmodul nicht mindestens zehn Studenten zugelassen, kann das Wahlpflichtmodul vom Modulangebot gestrichen werden. Auf schriftlichen Antrag kann der Student an Stelle eines Wahlpflichtmoduls für ein Zusatzmodul zugelassen werden. Über den Antrag entscheidet der Prüfungsausschuss. Ein Anspruch darauf, dass der Student zu einem bestimmten Wahlpflichtmodul zugelassen oder ihm ein bestimmtes Wahlpflichtmodul angeboten wird, besteht nicht.

(9) In der Regel im 6. Semester durchläuft der Student eine mindestens 12 Wochen dauernde Praxisphase (Praxisprojekt). Während der Dauer des Studiums hat der Student in einem Semester seiner Wahl an dem Veranstaltungszyklus des Studiums generelle teilzunehmen. Empfohlen wird dafür das 2. Semester.

## **§ 5 Studienberatung**

(1) Die allgemeine Studienberatung erfolgt durch das Dezernat Studienangelegenheiten der HTWK Leipzig. Sie erstreckt sich insbesondere auf Fragen der Studienmöglichkeiten, der Immatrikulation, Exmatrikulation und Beurlaubung sowie auf allgemeine studentische Angelegenheiten.

(2) Die studienbegleitende fachliche und organisatorische Beratung wird in Verantwortung der Fakultät durchgeführt. Sie umfasst insbesondere Fragen zu Modulhalten und zum Studienablauf. Im Rahmen vorhandener Kapazitäten finden, insbesondere zur Unterstützung von Studienanfängern, Tutorien statt.

(3) In prüfungsrechtlichen Angelegenheiten, insbesondere zum Vorgehen gegen belastende Entscheidungen der HTWK Leipzig, berät der Justitiar.

(4) Wer nicht spätestens in der Prüfungsperiode des 2. Semesters wenigstens einen Prüfungserstversuch unternommen hat, muss sich einer Beratung nach Absatz 2 Satz 1 unterziehen.

## **§ 6 Schlussbestimmungen**

(1) Die Studienordnung des Bachelorstudiengangs Informatik wurde am 29.05.2019 vom Fakultätsrat der Fakultät IM beschlossen und am 27.08.2019 durch das Rektorat genehmigt. Sie tritt ab 01.10.2019 in Kraft und gilt für alle eingeschriebenen Studierenden. Gleichzeitig treten alle vorhergehenden Studienordnungen des Bachelorstudiengangs INB der HTWK Leipzig außer Kraft.

(2) Glaubt ein Student, aus der vor dieser Studienordnung geltenden Studienordnung eine für sich günstigere Regelung herleiten zu können, kann er auf schriftlichen Antrag die Anwendung dieser Regel verlangen. Die Antragstellung ist bis spätestens 30.09.2020 möglich.

(3) Die Studienordnung des Studiengangs INB wird im Internetportal der HTWK Leipzig unter [www.htwk-leipzig.de](http://www.htwk-leipzig.de) veröffentlicht.

### **Anlagen**

- 1.) Modulhandbuch
- 2.) Praktikumsordnung

## Allgemein

<b>Studiengangskürzel</b>	18INB
<b>Studiengang</b>	Informatik   Bachelor Computer Science   Bachelor
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Abschluss</b>	Bachelor
<b>Erste Immatrikulation</b>	2018
<b>Status</b>	Prüfung Prorektorat Bildung positiv
<b>Regelstudienzeit in Semestern</b>	6 Semester
<b>Erforderliche Leistungspunkte</b>	180
<b>Studienmodus</b>	In Vollzeit studierbar
<b>Studienmodell</b>	Keine Angabe
<b>Für den Auslandsaufenthalt empfohlen</b>	-
<b>Studiengangsverantwortliche/r</b>	Prof. Dr. rer. nat. Karsten Weicker <a href="mailto:karsten.weicker@htwk-leipzig.de">karsten.weicker@htwk-leipzig.de</a>
<b>Ordnungen</b>	

## Studienablaufplan

Struktureinheit / Modul	Art	ECTS-Punkte	Semesterwochenstunden (V/S/Ü/P) / Prüfungen						
			1. Sem.	2. Sem.	3. Sem.	4. Sem.	5. Sem.	6. Sem.	
<b>Modellierung</b> Modelling C114 (INB1010, MIB1010, MI-BI1010)	Pflichtmodul	7	4/2/0/0  PVL PVL PK <sup>1</sup>						
<b>Mathematik für Informatiker I</b> Mathematics in Computer Science I N662 (INB1050, MIB1050, MIB-BI1050)	Pflichtmodul	8	4/0/2/0  PVL PK <sup>1</sup>						
<b>Anwendungsorientierte Programmierung</b> Applied Programming C680 (INB2029, MIB2029, MIB-BI2029)	Pflichtmodul	8	2/2/0/0  PVL PJ <sup>1</sup>	2/2/0/0  PVL PJ <sup>1</sup>					
<b>Technische Informatik I</b> Computer Engineering I C193 (INB2039)	Pflichtmodul	10	4/4/0/0  PVL PK <sup>1</sup> PVL PK <sup>1</sup>	0/0/0/1  PX <sup>1</sup>					
<b>Technische Informatik II</b> Computer Engineering II C421 (INB2049)	Pflichtmodul	6		3/2/1/0  PVL PK <sup>1</sup> PB <sup>1</sup> PP					



Struktureinheit / Modul	Art	ECTS-Punkte	Semesterwochenstunden (V/S/Ü/P) / Prüfungen					
			1. Sem.	2. Sem.	3. Sem.	4. Sem.	5. Sem.	6. Sem.
<b>Algorithmen und Datenstrukturen</b> Algorithms and Data Structures C300 (INB2050, MIB2050, MIB-BI2050)	Pflichtmodul	6		4/2/0/0  PVL PVL PK <sup>1</sup>				
<b>Mathematik für Informatiker II</b> Mathematics in Computer Science II N250 (INB2150)	Pflichtmodul	8		3/3/0/0  PVL PK <sup>1</sup>				
<b>Technisches Englisch und Schlüsselqualifikationen</b> Technical English and Key Qualifications C745 (INB3069, MIB3069, MIB-BI4069)	Pflichtmodul	7		1/1/2/2  PVL PVL PR PC PJ	1/0/0/0  PT <sup>2,3</sup>			
<b>Betriebssysteme und Rechnernetze</b> Operating Systems and Computer Networks C287 (INB3039, MIB3039, MIB-BI3039)	Pflichtmodul	6			4/0/0/2  PC <sup>1</sup> PVL PK <sup>1</sup>			
<b>Automaten und formale Sprachen</b> Automata and formal languages C993 (INB3010, MIB8310)	Pflichtmodul	5			2/2/0/0  PVL PVL PK <sup>1</sup>			

Struktureinheit / Modul	Art	ECTS-Punkte	Semesterwochenstunden (V/S/Ü/P) / Prüfungen					
			1. Sem.	2. Sem.	3. Sem.	4. Sem.	5. Sem.	6. Sem.
<b>Technische Informatik III</b> Introduction to Computer Systems III C705 (INB3049)	Pflichtmodul	6			2/2/0/1  PX PVL PK <sup>1</sup>			
<b>Datenbanken</b> Database Systems C719 (INB3050, MIB3050, MIB-BI3050)	Pflichtmodul	5			2/2/0/0  PVL PK <sup>1</sup>			
<b>Softwaretechnik</b> Software Engineering C559 (INB3070, MIB3070, MIB-BI3070)	Pflichtmodul	5			2/2/0/0  PVL PVL PK			
<b>Softwareprojekt</b> Software Engineering Project C732 (INB4080, MIB4080, MIB-BI4080)	Pflichtmodul	8			0.5/0/0/0.5	0.5/0/0/0.5  PJ		
<b>Fortgeschrittene Programmierung</b> Advanced Programming C393 (INB4010, MIB4010, MIB-BI4010)	Pflichtmodul	5				2/2/0/0  PVL PK <sup>1</sup>		
<b>IT-Sicherheit</b> IT Security C799 (INB5010, MIB5010, MIB-BI5010)	Pflichtmodul	5					2/2/0/0  PVL PK <sup>1</sup>	

Struktureinheit / Modul	Art	ECTS-Punkte	Semesterwochenstunden (V/S/Ü/P) / Prüfungen					
			1. Sem.	2. Sem.	3. Sem.	4. Sem.	5. Sem.	6. Sem.
<b>Einführung in die Betriebswirtschaftslehre</b> Introduction to Business Administration W861 (INB5060, MIB8130, MIB-BI8460)	Pflichtmodul	5					2/2/0/0  PVL PK	
<b>Praxisprojekt</b> Practical Project C222 (INB6000, MIB6000, MIB-BI6000)	Pflichtmodul	15						X  PVL PP
<b>Bachelormodul</b> Bachelor's Module C241 (INB/MIB9010)	Pflichtmodul	15						X  PH <sup>1</sup> PKQ <sup>1</sup>
<b>Wahlpflicht</b> Es sind mindestens 2 Bausteine zu bestehen.	Wahlpflichtbereich	40				16	16	
<b>Hardware-Entwurfstechnik</b> Hardware Design Tools C448 (INB8100)	Wahlpflichtmodul	5				2/2/0/0  PVL PM		
<b>Einführung in ERP-Software (SAP)</b> Introduction to ERP Software (SAP) N450 (INB8130)	Wahlpflichtmodul	5				2/0/2/0  PC		
<b>Diskrete Mathematik</b> Discrete Mathematics N915 (INB8160, MIB8160)	Wahlpflichtmodul	5				2/2/0/0  PVL PK <sup>1</sup>		

Struktureinheit / Modul	Art	ECTS-Punkte	Semesterwochenstunden (V/S/Ü/P) / Prüfungen					
			1. Sem.	2. Sem.	3. Sem.	4. Sem.	5. Sem.	6. Sem.
<b>Mobile Computing</b> Mobile Computing C652 (INB8490, MIB8490, MIB-BI8490)	Wahlpflichtmodul	5				2/2/0/0 PC		
<b>Grundlagen der mobilen Robotik</b> Principles of mobile robotics C010	Wahlpflichtmodul	5					2/0/2/0 PVL PH <sup>1</sup> PR <sup>1</sup>	
<b>Dokumentbeschreibungssprachen</b> Document Description Languages C651 (MIB-BI5040, INB8040, MIB8040)	Wahlpflichtmodul	5					2/2/0/0 PVL PJ	
<b>Numerische Mathematik</b> Numerical Mathematics N341 (INB8120)	Wahlpflichtmodul	5					2/2/0/0 PVL PK	
<b>Computeranimation</b> Computer Animation C182 (INB8140, MIB8140)	Wahlpflichtmodul	5					2/2/0/0 PJ	
Baustein Technologien für Softwaresysteme Es sind mind. 3 Module zu wählen.	Modulbereich	20				8	8	
<b>Grundlagen der Künstlichen Intelligenz</b> Foundations of Artificial Intelligence C622 (INB8012)	Wahlpflichtmodul	5				2/2/0/0 PVL PK		

Struktureinheit / Modul	Art	ECTS-Punkte	Semesterwochenstunden (V/S/Ü/P) / Prüfungen					
			1. Sem.	2. Sem.	3. Sem.	4. Sem.	5. Sem.	6. Sem.
<b>Computergrafik</b> Computer Graphics C121 (MIB4030, INB8013, MIB-BI8470)	Wahlpflichtmodul	5				2/2/0/0  PVL PK <sup>1</sup>		
<b>Audio-Video-Kommunikation</b> Audio-Video Communication C064 (INB8014)	Wahlpflichtmodul	5					0/0/2/2  PM	
<b>Datenbanken (Aufbaukurs)</b> Database Systems (advanced level) C720 (MIB-BI5050, INB8150, MIB8150)	Wahlpflichtmodul	5					2/2/0/0  PVL PM	
<b>Baustein Programmieretechniken</b> Es sind mind. 3 Module zu wählen.	Modulbereich	20				8	8	
<b>Multimediale Webprogrammierung</b> Multimedia Web Programming C741 (MIB4020, MIB-BI4020, INB8021)	Wahlpflichtmodul	5				2/2/0/0  PVL PK <sup>1</sup>		
<b>Assemblerprogrammierung</b> Assembler Programming C960 (INB8022)	Wahlpflichtmodul	5				2/2/0/0  PJ <sup>1</sup>		
<b>Sprachkonzepte der parallelen Programmierung</b> Language Concepts for Parallel Programming C529 (INB8023)	Wahlpflichtmodul	5					2/2/0/0  PVL PK <sup>1</sup>	

Struktureinheit / Modul	Art	ECTS-Punkte	Semesterwochenstunden (V/S/Ü/P) / Prüfungen					
			1. Sem.	2. Sem.	3. Sem.	4. Sem.	5. Sem.	6. Sem.
<b>Systemprogrammierung</b> Systems Programming C530 (INB8024)	Wahlpflichtmodul	5					2/0/0/2  PVL PM <sup>1</sup>	
Baustein Technische Systeme Es sind mind. 3 Module zu wählen.	Modulbereich	20				8	8	
<b>Prozessautomatisierung</b> Automation of Technical Processes C383 (INB8032)	Wahlpflichtmodul	5				2/0/0/2  PVL PM		
<b>Mikroprogrammierung und Mikroprozessoren</b> Microprogramming and Microprocessors C004 (INB8034)	Wahlpflichtmodul	5				2/0/0/2  PJ <sup>1</sup>		
<b>Grundlagen Internet-basierter Informationssysteme</b> Introduction to Internet Based Information Systems C247 (INB8031, MIB8031, MIB-BI8410)	Wahlpflichtmodul	5					2/0/2/0  PM <sup>1</sup>	
<b>Digitale Signalverarbeitung</b> Digital Signal Processing C807 (INB8033)	Wahlpflichtmodul	5					2/2/0/0  PM <sup>1</sup>	
Summe SWS pro Semester:			24	29	25	21	24	0
Summe ECTS-Credits pro Semester:			27	32	31	30	30	30

- <sup>1</sup> - Die Prüfungsleistung muss mit mindestens „ausreichend“ (4,0) bestanden sein.  
<sup>2</sup> - Nicht benotete Prüfungsleistung, die bestanden sein muss.  
<sup>3</sup> - Die Prüfungsleistung wird in englischer Sprache abgenommen.

PB - Prüfung Beleg  
PC - Prüfung Computerarbeit  
PH - Prüfung Hausarbeit  
PJ - Prüfung Projektarbeit  
PK - Prüfung Klausurarbeit  
PKQ - Prüfung Kolloquium  
PM - Prüfung mündliches Fachgespräch  
PP - Prüfung Präsentation  
PR - Prüfung Referat  
PT - Prüfung Testat  
PVL - Prüfungsvorleistung  
PX - Prüfung Experiment

<b>Modul</b>	Bachelormodul Bachelor's Module
<b>Modulnummer</b>	C241 [INB/MIB9010] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Sommersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. Karsten Weicker <a href="mailto:karsten.weicker@htwk-leipzig.de">karsten.weicker@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	
<b>Sprache(n)</b>	Deutsch in „Bachelorarbeit“  Deutsch in „Bachelorkolloquium“
<b>ECTS-Leistungspunkte</b>	15 ECTS-Punkte
<b>Workload</b>	450 Stunden 360 Stunden in „Bachelorarbeit“ 90 Stunden in „Bachelorkolloquium“
<b>Lehrveranstaltungen</b>	0 SWS 0 SWS in „Bachelorarbeit“ 0 SWS in „Bachelorkolloquium“
<b>Selbststudienzeit</b>	450 Stunden 360 Stunden in „Bachelorarbeit“ 90 Stunden in „Bachelorkolloquium“
<b>Prüfungsvorleistung(en)</b>	Keine
<b>Prüfungsleistung(en)</b>	Prüfung Hausarbeit Prüfungsdauer: 3 Monate   Wichtig: 75%   nicht kompensierbar in „Bachelorarbeit“  Prüfung Kolloquium Prüfungsdauer: 90 Minuten   Wichtig: 25%   nicht kompensierbar in „Bachelorkolloquium“



<b>Lehr- und Lernformen</b>	<p>„<b>Bachelorarbeit</b>“: keine Angabe</p> <p>„<b>Bachelorkolloquium</b>“: keine Angabe</p>
<b>Medienform</b>	<p>„<b>Bachelorarbeit</b>“: keine Angabe</p> <p>„<b>Bachelorkolloquium</b>“: keine Angabe</p>
<b>Lehrinhalte/Gliederung</b>	<p>„<b>Bachelorarbeit</b>“: themenspezifisch</p> <p>„<b>Bachelorkolloquium</b>“: themenspezifisch</p>
<b>Qualifikationsziele</b>	<p><b>Bachelorarbeit:</b> Mit der Bachelorarbeit zeigt der Student, dass er in der Lage ist, ein umfangreiches Problem seines Fachgebiets innerhalb einer vorgegebenen Frist mit üblichen fachspezifischen Methoden zu bearbeiten und dazu eine schriftliche wissenschaftliche Arbeit zu verfassen. Das Thema wird durch einen Professor (den Betreuer der Arbeit) festgelegt.</p> <p><b>Bachelorkolloquium:</b> Im Bachelorkolloquium stellt der Student die Fähigkeit unter Beweis, Inhalt, Methodik und Ergebnisse seiner Arbeit objektiv und ansprechend zu präsentieren und in der wissenschaftlichen Diskussion zu verteidigen</p>
<b>Zulassungsvoraussetzung</b>	Festlegung durch Prüfungsordnung
<b>Empfohlene Voraussetzungen</b>	Keine
<b>Literaturhinweise</b>	<p>„<b>Bachelorarbeit</b>“: themenspezifisch</p> <p>„<b>Bachelorkolloquium</b>“: themenspezifisch</p>
<b>Aktuelle Lehrressourcen</b>	<p>„<b>Bachelorarbeit</b>“: keine</p> <p>„<b>Bachelorkolloquium</b>“: keine</p>
<b>Hinweise</b>	Keine Angabe
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Betriebssysteme und Rechnernetze Operating Systems and Computer Networks
<b>Modulnummer</b>	C287 [INB3039, MIB3039, MIB-BI3039] Version: 2
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Wintersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr.-Ing. Jean-Alexander Müller <a href="mailto:jean-alexander.mueller@htwk-leipzig.de">jean-alexander.mueller@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr.-Ing. Jean-Alexander Müller <a href="mailto:jean-alexander.mueller@htwk-leipzig.de">jean-alexander.mueller@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch in "Betriebssysteme"  Deutsch in "Rechnernetze"
<b>ECTS-Leistungspunkte</b>	6 ECTS-Punkte
<b>Workload</b>	180 Stunden 120 Stunden in "Betriebssysteme" 60 Stunden in "Rechnernetze"
<b>Lehrveranstaltungen</b>	6 SWS (4 SWS Vorlesung   2 SWS Praktikum) 4 SWS (2 SWS Vorlesung   2 SWS Praktikum) in "Betriebssysteme" 2 SWS (2 SWS Vorlesung) in "Rechnernetze"
<b>Selbststudienzeit</b>	96 Stunden 34 Stunden Vorbereitung Lehrveranstaltung - Betriebssysteme 30 Stunden Selbststudium - Betriebssysteme 32 Stunden in "Rechnernetze"
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Beleg in "Rechnernetze"
<b>Prüfungsleistung(en)</b>	Prüfung Computerarbeit Prüfungsdauer: 4 Monate   Wichtigung: 50%   nicht kompensierbar in "Betriebssysteme"  Prüfung Klausurarbeit Prüfungsdauer: 90 Minuten   Wichtigung: 50%   nicht kompensierbar in "Rechnernetze"

<b>Lehr- und Lernformen</b>	<b>Betriebssysteme:</b> keine Angaben  <b>Rechnernetze:</b> keine Angaben
<b>Medienform</b>	<b>Betriebssysteme:</b> keine Angaben  <b>Rechnernetze:</b> keine Angaben
<b>Lehrinhalte/Gliederung</b>	<b>Betriebssysteme:</b> <ul style="list-style-type: none"> <li>• Aufgabenstellung und Begriffsbestimmung</li> <li>• Entwicklung von Rechnerarchitekturen und Betriebssystemen, Klassifikation</li> <li>• PC-Betriebssysteme als Beispiel</li> <li>• Prozesse, Dateisysteme, Nutzer</li> <li>• Kommandoprozeduren unter UNIX</li> <li>• parallele Prozesse unter UNIX</li> <li>• einfache Formen der Kommunikation paralleler Prozesse</li> <li>• praktische Übungen zur Programmierung von Kommandoprozeduren und parallelen Prozessen</li> </ul> <b>Rechnernetze:</b> <ul style="list-style-type: none"> <li>- Einführung in Netzwerktechnologien und Strukturen <ul style="list-style-type: none"> <li>- Datacenter / Vernetzung in Rechenzentren</li> <li>- Lokale Netze bis zum Intranet</li> <li>- Das Internet und andere Weitverkehrsnetze</li> <li>- Überblick zu Mobil- und Zugangsnetzen</li> </ul> </li> <li>- Architektur und Grundprinzipien <ul style="list-style-type: none"> <li>- Paketvermittlung, Referenzmodelle und Betriebsverfahren</li> <li>- Scheduling und Planung</li> <li>- Direktverbindungsnetze</li> <li>- Vermittlungsprinzipien, Routingverfahren</li> <li>- Tunnel, Overlay</li> <li>- Sicherheitsaspekte</li> </ul> </li> <li>- Technologien <ul style="list-style-type: none"> <li>- Internet Protocol (v4, v6, vX)</li> <li>- IEEE 802-Technologien</li> <li>- Virtualisierung, SDN, OpenFlow</li> <li>- Carrier Ethernet, GMPLS</li> </ul> </li> </ul>
<b>Qualifikationsziele</b>	<b>Betriebssysteme:</b> Die Studierenden können Grundkonzepte von modernen Betriebssystemen formal und sprachlich korrekt beschreiben und sind in der Lage, sie auf PC-Plattformen anzuwenden und nutzbar zu machen. Sie können selbständig und mit angemessenen Mitteln Betriebssysteme auf PC-Plattformen installieren und anpassen. Sowohl die Erstellung von Unix-spezifischen Anwendungsprogrammen unter Einsatz der Unix-API wie auch die Programmierung von Kommandoprozeduren kann selbständig unter Nutzung der vorhandenen Systemdokumentationen durchgeführt werden.  <b>Rechnernetze:</b> Es besteht detailliertes, anwendungsfähiges Fachwissen auf dem Gebiet der Netzwerktechnologien, Strukturen und deren Grundprinzipien. Aufsetzend auf dem Verständnis der Grundprinzipien sowie der erworbenen praktischen Fähigkeiten sind sie in der Lage veränderte Methoden und Trends zu erkennen und deren Potential gegenüber etablierten Technologien zu ermitteln.

<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Fertigkeiten in der Programmierung (derzeit C-Programmierung)
<b>Literaturhinweise</b>	<p><b>Betriebssysteme:</b>  A. S. Tanenbaum: „Moderne Betriebssysteme“, Pearson Verlag, 2003.  open SuSE: Linux Anwenderhandbuch und aktuelle Distribution.  R. Göstenmeier: „Das Einsteigerseminar Linux“, bhv-Taschenbuch, 2012.</p> <p><b>Rechnernetze:</b>  - P. L. Dordal: "<a href="#">An Introduction to Computer Networks</a>", CC BY-NC-ND 3.0, 2019.  - A. S. Tanenbaum, D. J. Wetherall: „Computer Networks“, Prentice Hall, 5. Auflage, 2010.  - K. R. Fall, W. R. Stevens: "TCP/IP Illustrated volume 1: The Protocols", Addison-Wesley, 2011.  - L. L. Peterson, B. S. Davie: "Computer Networks: A Systems Approach", Morgan Kaufmann, 5. Auflage, 2011.  - T. Nadeu, K. Gray: "SDN: Software Defined Networks", O'Reilly, 2013.</p>
<b>Aktuelle Lehrressourcen</b>	<p><b>Betriebssysteme:</b> keine Angaben</p> <p><b>Rechnernetze:</b> keine Angaben</p>
<b>Hinweise</b>	Keine Angabe
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Modellierung Modelling
<b>Modulnummer</b>	C114 [INB1010, MIB1010, MI-BI1010] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Wintersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. Sibylle Schwarz <a href="mailto:sibylle.schwarz@htwk-leipzig.de">sibylle.schwarz@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. Sibylle Schwarz <a href="mailto:sibylle.schwarz@htwk-leipzig.de">sibylle.schwarz@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	7 ECTS-Punkte
<b>Workload</b>	210 Stunden
<b>Lehrveranstaltungen</b>	6 SWS (4 SWS Vorlesung   2 SWS Seminar)
<b>Selbststudienzeit</b>	126 Stunden
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Beleg Prüfungsvorleistung Präsentation
<b>Prüfungsleistung(en)</b>	Prüfung Klausurarbeit Prüfungsdauer: 120 Minuten   Wichtigung: 100%   nicht kompensierbar
<b>Lehr- und Lernformen</b>	-
<b>Medienform</b>	keine Angabe

<b>Lehrinhalte/Gliederung</b>	<p>Modellierung und formale Darstellung von</p> <ul style="list-style-type: none"> <li>- Daten durch Mengen, Mengenoperationen</li> <li>- Zusammenhängen durch Relationen, Funktionen, Äquivalenz- Ordnungsrelationen, Graphen</li> <li>- strukturierten Daten durch Wörter, Texte, Sprachen, Bäume, Signaturen, Terme, strukturelle Induktion, algebraische Strukturen</li> <li>- Eigenschaften und Anforderungen in Logiken (jeweils Syntax, Semantik, Folgern, Schließen)</li> <li>- Software-Schnittstellen durch abstrakte Datentypen</li> <li>- Abläufen und Berechnungen durch Zustandsübergangssysteme jeweils mit praktischen Modellierungsbeispielen</li> </ul>
<b>Qualifikationsziele</b>	<p>Die Studierenden können mathematische und logische Grundkonzepte zur Modellierung praktischer Aufgabenstellungen anwenden.</p> <p>Sie können Anforderungen an Software und Systeme formal beschreiben und wissen, dass deren Korrektheit mit formalen Methoden nachweisbar ist.</p>
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Keine
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- U. Kastens, H. Kleine Büning: "Modellierung: Grundlagen und formale Methoden", Hanser, 2008.</li> <li>- M. Huth, M. Ryan: "Logic in Computer Science", Cambridge University Press, 2010.</li> <li>- U. Schöning: "Theoretische Informatik - kurzgefasst", Spektrum, in der aktuellen Auflage.</li> <li>- M. Broy, R. Steinbrüggen: "Modellbildung in der Informatik", Springer, 2004</li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	regelmäßiges erfolgreiches Lösen der praktischen Übungsaufgaben (PVB) und 3 Kurzvorträge zu schriftlichen Übungsaufgaben (PVP)
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Mathematik für Informatiker I Mathematics in Computer Science I
<b>Modulnummer</b>	N662 [INB1050, MIB1050, MIB-BI1050] Version: 1
<b>Fakultät</b>	MNZ(Ma): Mathematik - Mathematisch-Naturwissenschaftliches Zentrum
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Wintersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. habil. Hans-Jürgen Dobner <a href="mailto:hans-juergen.dobner@htwk-leipzig.de">hans-juergen.dobner@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. habil. Hans-Jürgen Dobner <a href="mailto:hans-juergen.dobner@htwk-leipzig.de">hans-juergen.dobner@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	8 ECTS-Punkte
<b>Workload</b>	240 Stunden
<b>Lehrveranstaltungen</b>	6 SWS (4 SWS Vorlesung   2 SWS Übung)
<b>Selbststudienzeit</b>	156 Stunden 60 Stunden Vorbereitung Lehrveranstaltung 32 Stunden Bearbeitung Prüfungsvorleistung 22 Stunden Vorbereitung Prüfung 42 Stunden Selbststudium
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Beleg
<b>Prüfungsleistung(en)</b>	Prüfung Klausurarbeit Modulprüfung   Prüfungsdauer: 120 Minuten   Wichtigkeit: 100%   nicht kompensierbar
<b>Lehr- und Lernformen</b>	Vorlesung: Tafel und Beamer, ein Lückenskript wird bereitgestellt  Seminare: Vertiefung der Vorlesungsinhalte durch eigenständiges Lösen von Aufgaben
<b>Medienform</b>	-

<b>Lehrinhalte/Gliederung</b>	Mengen, Aussagen, Beweistechniken, Algebraische Strukturen, Vektorräume, Basis und Dimension, Lineare Abbildungen und Matrizen, Lineare Gleichungssysteme. Ungleichungen, Folgen und Konvergenz, Stetigkeit, Grenzwertsätze, Reihen, Ableitung und Anwendungen der Differenzialrechnung
<b>Qualifikationsziele</b>	Nach erfolgreichem Abschluss des Moduls kennen die Studierenden die wichtigsten Konzepte, welche für die Informatik von Bedeutung sind. Hierzu gehört ein solides mathematisches Grundwissen über Mengen, Aufbau des Zahlensystems, Aussagen, Abbildungen und grundlegende Beweistechniken. Im Bereich der Algebra kennen die Studierenden die Vektorraumstruktur und wissen die geometrischen, arithmetischen sowie strukturbetont-abstrakten Aspekte Informatik-bezogen einsetzen. Die Studierenden beherrschen alle Gesichtspunkte der Vektorräume, wozu der sichere Umgang mit den zentralen Begriffen - Lineare Abhängigkeit/Unabhängigkeit, Basis, Dimension, Teilraumstrukturen und Lineare Abbildungen - zählt. Die Studierenden lernen mit Linearen Gleichungssystemen eine der wichtigsten Aufgaben der linearen Algebra kennen und eignen sich fundierte Kenntnisse zu deren Lösung und deren Einordnung in den Gesamtkomplex der Linearen Algebra an. Ferner haben die Studierenden ein tiefes Verständnis für den Zusammenhang zwischen Matrizen und linearen Abbildungen entwickelt. Im Bereich der Analysis lernen die Studierenden den Umgang mit Ungleichungen und Abschätzungen. Grundlage der Analysis ist das Beherrschen von Folgen und deren Konvergenzverhalten. Mit deren Anwendung im Rahmen der Analyse von Algorithmen werden Bezüge zur Informatik aufgezeigt. Mit Reihen lernen Studierende weitere (spezielle) Folgen kennen. Neben der Stetigkeit von Funktionen einer Veränderlichen wird das Studium elementarer Funktionen und deren Eigenschaften vermittelt. Mit der Ableitung und den wichtigsten Ableitungsregeln lernen die Studierenden ein wichtiges Werkzeug zur Untersuchung des Verhaltens von Funktionen kennen. Im Rahmen der Differenzialrechnung lernen die Studierenden Bedingungen für Extrema, die Regeln von de l'Hospital und die Approximation von Funktionen durch Taylor-Polynome kennen.
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Keine
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- Konkrete Mathematik (nicht nur) für Informatiker, Edmund Weitz, Springer Spektrum 2018</li> <li>- Höhere Mathematik in Rezepten, 2. Auflage, Christian Karpfinger, Springer Spektrum 2015</li> <li>- Arbeitsbuch Höhere Mathematik in Rezepten, Christian Karpfinger, Springer Spektrum 2014</li> <li>- Mathematik für Informatiker, Steffen Goebbels, Jochen Rethmann, Springer Vieweg 2014</li> <li>- Mathematik für Informatiker, 2. Auflage, Matthias Schubert, Springer Vieweg+Teubner 2012</li> <li>- Mathematik für Informatiker, Band 1, 4. Auflage, Gerald Teschl, Susanne Teschl, Springer Vieweg 2013</li> <li>- Mathematik für Informatiker, Band 2, 3. Auflage, Gerald Teschl, Susanne Teschl, Springer Vieweg 2014</li> <li>- Mathematik für Informatiker, 2. Auflage, Dirk Hachenberger, Pearson Studium 2008</li> <li>- Toolbox Mathematik für MINT-Studiengänge, Erhard Cramer, Udo Kamps, Jessica Lehmann, Sebastian Walcher, Springer Spektrum 2017</li> <li>- So einfach ist Mathematik – Zwölf Herausforderungen im ersten Semester, Dirk Langemann, Vanessa Sommer, Springer Spektrum 2017</li> <li>- Mathematik-Klausurtrainer, Reinhard Strehlow, Hanser 2007</li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	Keine Angabe



<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	
--	--

<b>Modul</b>	Anwendungsorientierte Programmierung Applied Programming
<b>Modulnummer</b>	C680 [INB2029, MIB2029, MIB-BI2029] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	2 Semester
<b>Turnus</b>	Wintersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. Heinrich Krämer <a href="mailto:heinrich.kraemer@htwk-leipzig.de">heinrich.kraemer@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. Mario Hlawitschka <a href="mailto:mario.hlawitschka@htwk-leipzig.de">mario.hlawitschka@htwk-leipzig.de</a> Dozent/-in in: "Anwendungsorientierte Programmierung I"  Prof. Dr. rer. nat. Heinrich Krämer <a href="mailto:heinrich.kraemer@htwk-leipzig.de">heinrich.kraemer@htwk-leipzig.de</a> Dozent/-in in: "Anwendungsorientierte Programmierung II"
<b>Sprache(n)</b>	Deutsch in "Anwendungsorientierte Programmierung I"  Deutsch in "Anwendungsorientierte Programmierung II"
<b>ECTS-Leistungspunkte</b>	8 ECTS-Punkte
<b>Workload</b>	240 Stunden 120 Stunden in "Anwendungsorientierte Programmierung I" 120 Stunden in "Anwendungsorientierte Programmierung II"
<b>Lehrveranstaltungen</b>	8 SWS (4 SWS Vorlesung   4 SWS Seminar) 4 SWS (2 SWS Vorlesung   2 SWS Seminar) in "Anwendungsorientierte Programmierung I" 4 SWS (2 SWS Vorlesung   2 SWS Seminar) in "Anwendungsorientierte Programmierung II"
<b>Selbststudienzeit</b>	128 Stunden 30 Stunden Bearbeitung Prüfungsleistung - Anwendungsorientierte Programmierung I 34 Stunden Vorbereitung Lehrveranstaltung - Anwendungsorientierte Programmierung I 30 Stunden Bearbeitung Prüfungsleistung - Anwendungsorientierte Programmierung II 34 Stunden Vorbereitung Lehrveranstaltung - Anwendungsorientierte Programmierung II
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Beleg in "Anwendungsorientierte Programmierung I"  Prüfungsvorleistung Beleg in "Anwendungsorientierte Programmierung II"

<b>Prüfungsleistung(en)</b>	<p>Prüfung Projektarbeit Prüfungsdauer: 2 Wochen   Wichtigung: 50%   nicht kompensierbar in "Anwendungsorientierte Programmierung I"</p> <p>Prüfung Projektarbeit Prüfungsdauer: 4 Wochen   Wichtigung: 50%   nicht kompensierbar in "Anwendungsorientierte Programmierung II"</p>
<b>Lehr- und Lernformen</b>	<p><b>Anwendungsorientierte Programmierung I:</b> -</p> <p><b>Anwendungsorientierte Programmierung II:</b> -</p>
<b>Medienform</b>	<p><b>Anwendungsorientierte Programmierung I:</b> keine Angabe</p> <p><b>Anwendungsorientierte Programmierung II:</b> keine Angabe</p>
<b>Lehrinhalte/Gliederung</b>	<p><b>Anwendungsorientierte Programmierung I:</b></p> <ul style="list-style-type: none"> <li>- Imperative Programmierung <ul style="list-style-type: none"> <li>- Kontrollstrukturen</li> <li>- Unterprogramme</li> </ul> </li> <li>- Objektorientiertes Programmieren <ul style="list-style-type: none"> <li>- Verwendung von objektorientierten Datenstrukturen Vererbung sowie Schnittstellen und Klassen als deren Implementierungen</li> <li>- Ausnahmebehandlung</li> <li>- Vererbung</li> </ul> </li> <li>- Grundlagen des Umgangs mit Dateien und Speicher</li> </ul> <p><b>Anwendungsorientierte Programmierung II:</b></p> <ul style="list-style-type: none"> <li>- Objektorientiertes Programmieren <ul style="list-style-type: none"> <li>- Vererbung sowie Schnittstellen und Klassen als deren Implementierungen</li> <li>- Ausnahmebehandlung</li> <li>- Anwendung von generischen Datentypen, z.B. durch Arbeit mit dem Java Collection Framework</li> <li>- Einführung in die Gestaltung von graphischen Benutzeroberflächen</li> </ul> </li> </ul>
<b>Qualifikationsziele</b>	<p>Die Studenten kennen und verstehen Syntax und Semantik der Programmiersprachen C++ und Java. Sie sind in der Lage, formale und textuelle Beschreibungen von einfachen Algorithmen in kleine Programme gemäß des imperativen und objektorientierten Programmierparadigmas umzusetzen sowie einfache Probleme eigenständig zu lösen. Sie kennen Grundlagen der Objektorientiertheit, können Objekte identifizieren und als Klassen implementieren.</p>
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Keine

<b>Literaturhinweise</b>	<b>Anwendungsorientierte Programmierung I:</b> - U. Breyman: „Der C++ Programmierer“, Hanser, 2015. - B. Stroustrup: „Die C++ Programmiersprache“, Hanser, 2015.  <b>Anwendungsorientierte Programmierung II:</b> - C. Ullenboom: „Java ist auch nur eine Insel“, Galileo Computing, in der aktuellen Auflage. - J. Gosling et al. : „The Java™ Language Specification“, <a href="http://docs.oracle.com/javase/specs">http://docs.oracle.com/javase/specs</a>
<b>Aktuelle Lehrressourcen</b>	<b>Anwendungsorientierte Programmierung I:</b> keine  <b>Anwendungsorientierte Programmierung II:</b> keine
<b>Hinweise</b>	<b>Anwendungsorientierte Programmierung II:</b> Lehreinheit I: Belege (PVB): selbständig erarbeitete Programme (Belege). Lehreinheit II:Belege (PVB): Zwei selbständig erarbeitete Programme (Belege). Die Abnahme und Diskussion erfolgt in jeweils einem Seminar
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Technische Informatik I Computer Engineering I
<b>Modulnummer</b>	C193 [INB2039] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	2 Semester
<b>Turnus</b>	Wintersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr.-Ing. Axel Schneider <a href="mailto:axel.schneider@htwk-leipzig.de">axel.schneider@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	<p>Prof. Dr.-Ing. Axel Schneider <a href="mailto:axel.schneider@htwk-leipzig.de">axel.schneider@htwk-leipzig.de</a> Dozent/-in in: "Digitaltechnik I"</p> <p>Prof. Dr. rer. nat. Hanna Brodowsky <a href="mailto:hanna.brodowsky@htwk-leipzig.de">hanna.brodowsky@htwk-leipzig.de</a> Dozent/-in in: "Physik für Informatiker"</p> <p>Prof. Dr. rer. nat. Jens Wagner <a href="mailto:jens.wagner@htwk-leipzig.de">jens.wagner@htwk-leipzig.de</a> Dozent/-in in: "Hardwarepraktikum I"</p>
<b>Sprache(n)</b>	<p>Deutsch in "Digitaltechnik I"</p> <p>Deutsch in "Physik für Informatiker"</p> <p>Deutsch in "Hardwarepraktikum I"</p>
<b>ECTS-Leistungspunkte</b>	10 ECTS-Punkte
<b>Workload</b>	<p>300 Stunden</p> <p>120 Stunden in "Digitaltechnik I"</p> <p>120 Stunden in "Physik für Informatiker"</p> <p>60 Stunden in "Hardwarepraktikum I"</p>
<b>Lehrveranstaltungen</b>	<p>9 SWS (4 SWS Vorlesung   1 SWS Praktikum   4 SWS Seminar)</p> <p>4 SWS (2 SWS Vorlesung   2 SWS Seminar) in "Digitaltechnik I"</p> <p>4 SWS (2 SWS Vorlesung   2 SWS Seminar) in "Physik für Informatiker"</p> <p>1 SWS (1 SWS Praktikum) in "Hardwarepraktikum I"</p>
<b>Selbststudienzeit</b>	<p>172 Stunden</p> <p>64 Stunden in "Digitaltechnik I"</p> <p>64 Stunden in "Physik für Informatiker"</p> <p>44 Stunden in "Hardwarepraktikum I"</p>

<p><b>Prüfungsvorleistung(en)</b></p>	<p>Prüfungsvorleistung Beleg in "Digitaltechnik I"</p> <p>Prüfungsvorleistung Testat in "Physik für Informatiker"</p>
<p><b>Prüfungsleistung(en)</b></p>	<p>Prüfung Klausurarbeit Prüfungsdauer: 120 Minuten   Wichtung: 40%   nicht kompensierbar in "Digitaltechnik I"</p> <p>Prüfung Klausurarbeit Prüfungsdauer: 120 Minuten   Wichtung: 40%   nicht kompensierbar in "Physik für Informatiker"</p> <p>Prüfung Experiment Prüfungsdauer: 3 Monate   Wichtung: 20%   nicht kompensierbar in "Hardwarepraktikum I"</p>
<p><b>Lehr- und Lernformen</b></p>	<p><b>Digitaltechnik I:</b> keine Angabe</p> <p><b>Physik für Informatiker:</b> keine Angabe</p> <p><b>Hardwarepraktikum I:</b> keine Angabe</p>
<p><b>Medienform</b></p>	<p><b>Digitaltechnik I:</b> keine Angabe</p> <p><b>Physik für Informatiker:</b> keine Angabe</p> <p><b>Hardwarepraktikum I:</b> keine Angabe</p>
<p><b>Lehrinhalte/Gliederung</b></p>	<p><b>Digitaltechnik I:</b></p> <ul style="list-style-type: none"> <li>- Schaltalgebra</li> <li>- Synthese und Analyse von Schaltnetzen</li> <li>- Realisierung spezieller Schaltnetze</li> </ul> <p><b>Physik für Informatiker:</b></p> <ul style="list-style-type: none"> <li>- Elektrische und magnetische Felder</li> <li>- Lineare Netzwerke</li> <li>- Funktionsweise von Halbleiterbauelementen</li> <li>- Anlogschaltungen mit Halbleiterbauelementen</li> <li>- Logikschaltungen</li> </ul> <p><b>Hardwarepraktikum I:</b></p> <ul style="list-style-type: none"> <li>- Analoge und digitale Messtechnik</li> <li>- Kennlinien von Universal-Dioden, LED, Suppressordioden</li> <li>- Transistor als Schalter</li> <li>- Signalausbreitung auf Kabeln</li> </ul>

<b>Qualifikationsziele</b>	<p><b>Lehreinheit „Digitaltechnik I“:</b> Nach erfolgreichem Abschluss des Moduls sind die Studierenden in der Lage, schaltalgebraische Beschreibungsmethoden für unterschiedliche technische Aufgabenstellungen anzuwenden. Sie können durch ihr Wissen mittels verschiedener Methoden und Verfahren Schaltnetze selbsttätig entwerfen, optimieren und technisch umsetzen.</p> <p><b>Lehreinheit „Physik für Informatiker“:</b> Studenten sind in der Lage, vorgegebene elektronische Schaltungen wie z.B. passive Netzwerke und Verstärker in ihren Eigenschaften zu beurteilen und zu berechnen. Zur Lösung von Aufgabenstellungen können analoge und digitale Grundsaltungen konzipiert und dimensioniert werden. Zur Realisierung verwendete Bauelemente können nach ihren Eigenschaften auf Eignung beurteilt werden.</p> <p><b>Lehreinheit „Hardwarepraktikum I“:</b> Die Studenten haben ein grundsätzliches Verständnis für die Funktionen passiver und aktiver Bauelemente sowie digitaler Schaltkreise und können mit geeigneten Messmitteln deren Eigenschaften darstellen und bewerten. Die problembezogene Auswahl und Anwendung von Verfahren der computergestützten Messtechnik und von Messmitteln wie Multimeter und Oszilloskop wird von ihnen bei typischen Standardaufgaben beherrscht. Die Studierenden können komplexe Aufgabenstellungen analysieren und Lösungsabläufe planen und ausführen.</p>
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	<p>Fähigkeit zum logischen und algorithmischen Denken.  Geübter Umgang mit den physikalischen Grundgrößen und ihren Maßeinheiten sowie ihre Anwendung auf Gleichstromkreise.  Aus verbalen Aufgabenstellungen heraus können Gleichungen und Gleichungssysteme aufgestellt und mit den Methoden der Arithmetik gelöst werden. Vertrautheit mit Methoden der Infinitesimalrechnung zur Diskussion von Funktionen einer Variablen, Bestimmung von Flächen und Volumina.  Gerichtete Größen können mit den Methoden der Vektorrechnung behandelt werden.</p>
<b>Literaturhinweise</b>	<p><b>Digitaltechnik I:</b></p> <ul style="list-style-type: none"> <li>- K. Fricke: „Digitaltechnik“, Vieweg, in der aktuellen Auflage.</li> <li>- G. Scarbata: „Synthese und Analyse Digitaler Schaltungen“, Oldenbourg, in der aktuellen Auflage.</li> <li>- C. Siemers und A. Sikora: „Taschenbuch Digitaltechnik“, Fachbuchverlag Leipzig, in der aktuellen Auflage</li> </ul> <p><b>Physik für Informatiker:</b></p> <ul style="list-style-type: none"> <li>- K. Lüders: „Lehrbuchmanuskript“ (online verfügbar)</li> <li>- H. Lindner: „Physik für Ingenieure“, Vieweg, in der aktuellen Auflage.</li> <li>- G. Koß, W. Reinhold: „Lehr- und Übungsbuch Elektronik“, Fachbuchverlag Leipzig, in der aktuellen Auflage.</li> <li>- P. Reinhold: „Elektrotechnik für Informatiker“, Teubner, in der aktuellen Auflage.</li> <li>- J. Rybach: „Physik für Bachelors“, Fachbuchverlag Leipzig, in der aktuellen Auflage.</li> </ul> <p><b>Hardwarepraktikum I:</b></p> <ul style="list-style-type: none"> <li>- Aufgabenspezifische Versuchsanleitungen zu den Praktikumsversuchen</li> <li>- J. Hofmann: „Taschenbuch der Messtechnik“, Fachbuchverlag Leipzig, in der aktuellen Auflage.</li> </ul>

<b>Aktuelle Lehrressourcen</b>	<b>Digitaltechnik I:</b> keine  <b>Physik für Informatiker:</b> keine  <b>Hardwarepraktikum I:</b> keine
<b>Hinweise</b>	<b>Digitaltechnik I:</b> Belege (PVB): Es werden 4 Belege ausgereicht. Dabei müssen mindestens 50% der Punkte der Gesamtbelegleistung erreicht werden.  <b>Physik für Informatiker:</b> Testat (PVT): wöchentliche Aufgaben mit wöchentlichen schriftlichen Kurzkontrollen
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	



<b>Modul</b>	Technische Informatik II Computer Engineering II
<b>Modulnummer</b>	C421 [INB2049] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Sommersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr.-Ing. Axel Schneider <a href="mailto:axel.schneider@htwk-leipzig.de">axel.schneider@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr.-Ing. Axel Schneider <a href="mailto:axel.schneider@htwk-leipzig.de">axel.schneider@htwk-leipzig.de</a> Dozent/-in in: "Digitaltechnik II"  Prof. Dr. rer. nat. Jens Wagner <a href="mailto:jens.wagner@htwk-leipzig.de">jens.wagner@htwk-leipzig.de</a> Dozent/-in in: "Systemnahe Programmierung"
<b>Sprache(n)</b>	Deutsch in "Digitaltechnik II"  Deutsch in "Systemnahe Programmierung"
<b>ECTS-Leistungspunkte</b>	6 ECTS-Punkte
<b>Workload</b>	180 Stunden 120 Stunden in "Digitaltechnik II" 60 Stunden in "Systemnahe Programmierung"
<b>Lehrveranstaltungen</b>	6 SWS (3 SWS Vorlesung   1 SWS Übung   2 SWS Seminar) 4 SWS (2 SWS Vorlesung   2 SWS Seminar) in "Digitaltechnik II" 2 SWS (1 SWS Vorlesung   1 SWS Übung) in "Systemnahe Programmierung"
<b>Selbststudienzeit</b>	90 Stunden 60 Stunden in "Digitaltechnik II" 30 Stunden in "Systemnahe Programmierung"
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Beleg in "Digitaltechnik II"

<b>Prüfungsleistung(en)</b>	<p><b>Prüfung Klausurarbeit</b> Prüfungsdauer: 120 Minuten   Wichtung: 66.67%   nicht kompensierbar in "Digitaltechnik II"</p> <p><b>Prüfung Beleg</b> Prüfungsdauer: 4 Wochen   Wichtung: 16.67%   nicht kompensierbar in "Systemnahe Programmierung"</p> <p><b>Prüfung Präsentation</b> Prüfungsdauer: 15 Minuten   Wichtung: 16.67% in "Systemnahe Programmierung"</p>
<b>Lehr- und Lernformen</b>	<p><b>Digitaltechnik II:</b> keine Angabe</p> <p><b>Systemnahe Programmierung:</b> keine Angabe</p>
<b>Medienform</b>	<p><b>Digitaltechnik II:</b> keine Angabe</p> <p><b>Systemnahe Programmierung:</b> keine Angabe</p>
<b>Lehrinhalte/Gliederung</b>	<p><b>Digitaltechnik II:</b></p> <ul style="list-style-type: none"> <li>- Theoretische Grundlagen der Schaltwerke</li> <li>- Synthese von Schaltwerken</li> <li>- Analyse von Schaltwerken</li> <li>- Realisierung spezieller Schaltwerke</li> <li>- Grundlagen der Informations- und Codierungstheorie</li> </ul> <p><b>Systemnahe Programmierung:</b></p> <ul style="list-style-type: none"> <li>- Einführung mit historischer Rechentechnik</li> <li>- Mikroprozessoren und Mikroprozessorsysteme</li> <li>- Programmiermodell und Instruktionen</li> <li>- Programmieren ganzzahliger Arithmetik</li> <li>- Werkzeuge der Maschinenprogrammierung</li> </ul>
<b>Qualifikationsziele</b>	<p><b>Digitaltechnik II:</b> Nach erfolgreichem Abschluss des Moduls sind die Studierenden in der Lage, einerseits beliebige Schaltwerke bis zu einem bestimmten Komplexitätsgrad zu entwerfen und zu analysieren und andererseits die wichtigsten Standard-Schaltwerke hinsichtlich ihrer Funktionsweise zu interpretieren. Zusammenhänge zu angrenzenden Gebieten der Informatik werden dabei verdeutlicht und führen zu vertieften Kenntnissen über informationsverarbeitende Systeme aus Sicht der Hardware.</p> <p><b>Systemnahe Programmierung:</b> Die Studierenden sind in der Lage, Programmiermodell und Ausführungslogik von Mikroprozessoren zu beschreiben und die Ausdrucksmittel dieser Architekturen zur Lösung systemnaher Aufgabenstellungen adäquat einzusetzen. Algorithmen der Ganzzahlarithmetik und zur Manipulation von Datenstrukturen können auf die Systemarchitektur abgebildet und mittels einer einfachen Entwicklungsumgebung implementiert werden.</p>
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Theoretische und physikalische Grundlagen der Informatik, Fähigkeit zum Entwurf von Schaltnetzen, praktische Erfahrungen mit einer anwendungsorientierten Programmiersprache

<b>Literaturhinweise</b>	<b>Digitaltechnik II:</b> - K. Fricke: „Digitaltechnik“, Vieweg, in der aktuellen Auflage. - G. Scarbata: „Synthese und Analyse Digitaler Schaltungen“, Oldenbourg, in der aktuellen Auflage. - W. Dankmeier: „Codierung“, Vieweg, in der aktuellen Auflage. - C. Siemers und A. Sikora: „Taschenbuch Digitaltechnik“, Fachbuchverlag Leipzig, in der aktuellen Auflage.  <b>Systemnahe Programmierung:</b> - Gebhardt, A.: SIM8008, Entwicklungsumgebung für einen 8-Bit-Mikrocomputer.
<b>Aktuelle Lehrressourcen</b>	<b>Digitaltechnik II:</b> keine  <b>Systemnahe Programmierung:</b> keine
<b>Hinweise</b>	<b>Digitaltechnik II:</b> Belege (PVB): Es werden 4 Belege ausgereicht. Dabei müssen mindestens 50% der Punkte der Gesamtbelegleistung erreicht werden.
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Algorithmen und Datenstrukturen Algorithms and Data Structures
<b>Modulnummer</b>	C300 [INB2050, MIB2050, MIB-BI2050] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Sommersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. Karsten Weicker <a href="mailto:karsten.weicker@htwk-leipzig.de">karsten.weicker@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. Karsten Weicker <a href="mailto:karsten.weicker@htwk-leipzig.de">karsten.weicker@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	6 ECTS-Punkte
<b>Workload</b>	180 Stunden
<b>Lehrveranstaltungen</b>	6 SWS (4 SWS Vorlesung   2 SWS Seminar)
<b>Selbststudienzeit</b>	96 Stunden
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Beleg Prüfungsvorleistung Präsentation
<b>Prüfungsleistung(en)</b>	Prüfung Klausurarbeit Modulprüfung   Prüfungsdauer: 120 Minuten   Wichtigung: 100%   nicht kompensierbar
<b>Lehr- und Lernformen</b>	-
<b>Medienform</b>	keine Angabe
<b>Lehrinhalte/Gliederung</b>	<ul style="list-style-type: none"> <li>- Grundlagen</li> <li>- Einfache Suchalgorithmen (Listen und Felder)</li> <li>- Bäume (Suchbäume, AVL-Bäume, optimale Suchbäume)</li> <li>- Sortieren (Quicksort, Heapsort, Mergesort)</li> <li>- Hashing (extern, offen, Brent's Algorithmus)</li> <li>- Graphenalgorithmen (minimaler Spannbaum, kürzeste Wege, Rundreiseproblem)</li> <li>- Entwurfparadigmen: Divide-and-Conquer, dynamisches Programmieren, Backtracking, Greedy</li> </ul>

<b>Qualifikationsziele</b>	Nach erfolgreichem Abschluss des Moduls, haben die Studierenden die behandelten Standarddatenstrukturen und -algorithmen so weit verstanden, dass sie diese am Beispiel nachvollziehen können. Ferner können sie einfache Algorithmen bzgl. der Laufzeit und des Speicherbedarfs analysieren - u.a. unter Verwendung eines Mastertheorems. Algorithmen können in einem Anwendungsszenario implementiert werden. Laufzeitmessungen können theoretischen Resultaten gegenübergestellt werden. Für einfache Aufgabenstellungen können die Studierenden eigene Algorithmen entwickeln.
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Keine
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- K. Weicker, N. Weicker: "Algorithmen und Datenstrukturen", SpringerVieweg, 2013.</li> <li>- T. Ottmann, P. Widmayer: "Algorithmen und Datenstrukturen", Spektrum, in der aktuellen Auflage.</li> <li>- T. H. Cormen et al.: "Algorithmen - Eine Einführung", Oldenbourg, in der aktuellen Auflage.</li> <li>- R. Sedgewick: "Algorithmen in Java", Addison-Wesley, in der aktuellen Auflage.</li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	Belege (PVB), Präsentationen (PVP): wöchentliche Aufgaben mit Präsentation der Lösung an der Tafel (in kooperativen Gruppen), Programmieraufgaben. Jeweils 70% der Aufgaben müssen erfolgreich bearbeitet werden.
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Mathematik für Informatiker II Mathematics in Computer Science II
<b>Modulnummer</b>	N250 [INB2150] Version: 1
<b>Fakultät</b>	MNZ(Ma): Mathematik - Mathematisch-Naturwissenschaftliches Zentrum
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Sommersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. habil. Hans-Jürgen Dobner <a href="mailto:hans-juergen.dobner@htwk-leipzig.de">hans-juergen.dobner@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. habil. Hans-Jürgen Dobner <a href="mailto:hans-juergen.dobner@htwk-leipzig.de">hans-juergen.dobner@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	8 ECTS-Punkte
<b>Workload</b>	240 Stunden
<b>Lehrveranstaltungen</b>	6 SWS (3 SWS Vorlesung   3 SWS Seminar)
<b>Selbststudienzeit</b>	156 Stunden
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Beleg
<b>Prüfungsleistung(en)</b>	Prüfung Klausurarbeit Modulprüfung   Prüfungsdauer: 135 Minuten   Wichtigung: 100%   nicht kompensierbar
<b>Lehr- und Lernformen</b>	-
<b>Medienform</b>	keine Angabe

<b>Lehrinhalte/Gliederung</b>	<p><b>Methoden der Analysis und Linearen Algebra:</b>  Norm, Skalarprodukt, Determinanten, Bestimmtes und unbestimmtes Integral, Integrationsmethoden, Hauptsatz der Differenzial- und Integralrechnung, uneigentliches Integral, Fourier-Reihen, Funktionen mehrerer Veränderlicher, Kurven, partielle Ableitungen, Gebietsintegral, Substitution des Gebietsintegrals, Definitheit von Matrizen und Extrema bei Funktionen mehrerer Veränderlicher.</p> <p><b>Wahrscheinlichkeitsrechnung:</b>  Zufällige Versuche, Ereignisse, relative Häufigkeiten, Begriff der Wahrscheinlichkeit, bedingte Wahrscheinlichkeit, totale Wahrscheinlichkeit, Satz von Bayes, Unabhängigkeit von Ereignissen, Zufallsgrößen, Wahrscheinlichkeitsverteilung, spezielle diskrete und stetige Verteilungen, Kennwerte von Zufallsgrößen.</p>
<b>Qualifikationsziele</b>	<p><b>Methoden der Analysis und Linearen Algebra:</b>  Mit der Einführung der Determinanten und Eigenwerte verfügen die Studierenden über weitere Möglichkeiten zur Charakterisierung von Matrizen und linearen Abbildungen. Mit der Betrachtung von Potenzreihen lernen Studierende Darstellungsmöglichkeiten elementarer Funktionen und Möglichkeiten zur deren Darstellung auf Rechnern kennen. Der Begriff des bestimmten Integrals wird geometrisch motiviert; die Verbindung zwischen Integral- zur Differenzialrechnung wird aufgezeigt.  Die Studierenden beherrschen die wichtigsten Methoden zur Bestimmung bestimmter und unbestimmter Integrale. Im Rahmen der Integralrechnung werden auch uneigentliche Integrale behandelt. Mit der Fourier-Analyse lernen Studierende ein wichtiges Anwendungsgebiet der Integralrechnung kennen. Mit der Übertragung der Grundbegriffe (Konvergenz, Stetigkeit, Ableitung, Integral) auf Funktionen mehrerer Veränderlicher und exemplarischen Anwendungen erwerben die Studierenden ein tieferes Verständnis für das Zusammenspiel mathematischer Methoden aus Analysis und Algebra in der Informatik.</p> <p><b>Wahrscheinlichkeitsrechnung:</b>  Das Hauptaugenmerk besteht in der Vermittlung mathematischer Methoden zur Beschreibung und Untersuchung zufallsabhängiger Phänomene. Nach erfolgreichem Abschluss beherrscht der Student wahrscheinlichkeitstheoretische Grundbegriffe und Denkweisen. Hierdurch wird er insbesondere in die Lage versetzt, weitere Kenntnisse auf dem Gebiet der Wahrscheinlichkeitstheorie zu erwerben, die es ermöglichen, praktische Probleme zu lösen.</p>
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Mathematik für Informatiker I
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- O. Bretscher: „Linear Algebra with Applications“, Prentice Hall, 2009, 4. Auflage.</li> <li>- M. Brill: „Mathematik für Informatiker“, Hanser, 2005, 2. Auflage.</li> <li>- H.-J. Dobner, G. Dobner: „Lineare Algebra“, Elsevier, 2007.</li> <li>- H.-J. Dobner, B. Engelmann: „Analysis II“, Hanser, 2013, 2. Auflage.</li> <li>- D. Hachenberger: „Mathematik für Informatiker“, Pearson, 2008, 2. Auflage.</li> <li>- B. Thomas, M.D. Weir: „Analysis 2“, Pearson, 2014, 12. Auflage.</li> <li>- O. Beyer, H. Hackel, V. Pieper, J. Tiedge: „Wahrscheinlichkeitsrechnung und Mathematische Statistik“, Teubner, 1999, 8. Auflage.</li> <li>- K. Bosch: „Elementare Einführung in die Wahrscheinlichkeitsrechnung“, Vieweg+Teubner, 2011, 11. Auflage.</li> <li>- E. Cramer, U. Kamps: „Grundlagen der Wahrscheinlichkeitsrechnung und Statistik“, Springer, 2008, 2. Auflage.</li> <li>- Ch. Hesse: „Angewandte Wahrscheinlichkeitstheorie“, Vieweg+Teubner, 2003.</li> <li>- U. Krengel: „Einführung in die Wahrscheinlichkeitstheorie und Statistik“, Vieweg, 2005, 8. Auflage.</li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine

<b>Hinweise</b>	Keine Angabe
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	



<b>Modul</b>	Automaten und formale Sprachen Automata and formal languages
<b>Modulnummer</b>	C993 [INB3010, MIB8310] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Wintersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. Sibylle Schwarz <a href="mailto:sibylle.schwarz@htwk-leipzig.de">sibylle.schwarz@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. Sibylle Schwarz <a href="mailto:sibylle.schwarz@htwk-leipzig.de">sibylle.schwarz@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	5 ECTS-Punkte
<b>Workload</b>	150 Stunden
<b>Lehrveranstaltungen</b>	4 SWS (2 SWS Vorlesung   2 SWS Seminar)
<b>Selbststudienzeit</b>	94 Stunden
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Beleg Prüfungsvorleistung Präsentation
<b>Prüfungsleistung(en)</b>	Prüfung Klausurarbeit Modulprüfung   Prüfungsdauer: 90 Minuten   Wichtung: 100%   nicht kompensierbar
<b>Lehr- und Lernformen</b>	-
<b>Medienform</b>	keine Angabe
<b>Lehrinhalte/Gliederung</b>	Formale Sprachen und verschiedene Darstellungsformen dafür, reguläre Ausdrücke Grammatiken (Chomsky-Hierarchie, Pumping Lemmata)  Berechnungsmodelle: endliche Automaten, Kellerautomaten, Turingmaschinen  Ausblick auf Grenzen der Berechenbarkeit

<b>Qualifikationsziele</b>	Die Studierenden sind in der Lage, wichtige Klassen formaler Sprachen als Grundlage von Programmier- und Beschreibungssprachen einzuordnen und kennen die wesentlichen Eigenschaften der Sprachklassen. Sie kennen die entsprechenden abstrakten Maschinenmodelle und Algorithmen und können sie zur Darstellung und Lösung praktischer Aufgabenstellungen einsetzen. Die Studierenden wissen, dass nicht jedes formal darstellbare Problem algorithmisch lösbar ist.
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	anwendungsbereite Kenntnisse auf den Gebieten Modellierung, Logik, Algorithmen und Datenstrukturen, Aufwandsabschätzungen
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- J. E. Hopcroft, J. D. Ullman: "Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie", Addison-Wesley, aktuelle Auflage.</li> <li>- U. Schöning: "Theoretische Informatik - kurzgefasst", Spektrum, aktuelle Auflage.</li> <li>- D. Hoffmann: "Theoretische Informatik", Hanser, 2009.</li> <li>- R. Socher: "Theoretische Grundlagen der Informatik", Hanser, 2008</li> <li>- G. Vossen, K.-U. Witt: "Grundkurs Theoretische Informatik", Springer Vieweg, aktuelle Auflage.</li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	regelmäßiges erfolgreiches Lösen der praktischen Übungsaufgaben (PVB) und 3 Kurzvorträge zu schriftlichen Übungsaufgaben (PVP)
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Technische Informatik III Introduction to Computer Systems III
<b>Modulnummer</b>	C705 [INB3049] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Wintersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr.-Ing. Axel Schneider <a href="mailto:axel.schneider@htwk-leipzig.de">axel.schneider@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. Jens Wagner <a href="mailto:jens.wagner@htwk-leipzig.de">jens.wagner@htwk-leipzig.de</a> Dozent/-in in: "Hardwarepraktikum II"  Prof. Dr.-Ing. Axel Schneider <a href="mailto:axel.schneider@htwk-leipzig.de">axel.schneider@htwk-leipzig.de</a> Dozent/-in in: "Rechnerarchitektur"
<b>Sprache(n)</b>	Deutsch in "Hardwarepraktikum II"  Deutsch in "Rechnerarchitektur"
<b>ECTS-Leistungspunkte</b>	6 ECTS-Punkte
<b>Workload</b>	180 Stunden 60 Stunden in "Hardwarepraktikum II" 120 Stunden in "Rechnerarchitektur"
<b>Lehrveranstaltungen</b>	5 SWS (2 SWS Vorlesung   1 SWS Praktikum   2 SWS Seminar) 1 SWS (1 SWS Praktikum) in "Hardwarepraktikum II" 4 SWS (2 SWS Vorlesung   2 SWS Seminar) in "Rechnerarchitektur"
<b>Selbststudienzeit</b>	108 Stunden 44 Stunden in "Hardwarepraktikum II" 64 Stunden in "Rechnerarchitektur"
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Referat in "Rechnerarchitektur"

<b>Prüfungsleistung(en)</b>	<p><b>Prüfung Experiment</b>  Prüfungsdauer: 3 Monate   Wichtung: 33.33%  in "Hardwarepraktikum II"</p> <p><b>Prüfung Klausurarbeit</b>  Prüfungsdauer: 120 Minuten   Wichtung: 66.67%   nicht kompensierbar  in "Rechnerarchitektur"</p>
<b>Lehr- und Lernformen</b>	<p><b>Hardwarepraktikum II:</b>  keine Angabe</p> <p><b>Rechnerarchitektur:</b>  keine Angabe</p>
<b>Medienform</b>	<p><b>Hardwarepraktikum II:</b>  keine Angabe</p> <p><b>Rechnerarchitektur:</b>  keine Angabe</p>
<b>Lehrinhalte/Gliederung</b>	<p><b>Hardwarepraktikum II:</b></p> <ul style="list-style-type: none"> <li>- Kombinatorische Logik und Flipflops</li> <li>- Mikrocontroller in Steuerungsanwendungen</li> <li>- Schnittstellen und Kommunikation</li> <li>- Automatenentwurf in einer HDL</li> </ul> <p><b>Rechnerarchitektur:</b></p> <ul style="list-style-type: none"> <li>- Grundlagen der Rechnerarchitektur</li> <li>- Prozessortypen und Befehlssätze</li> <li>- Leistungsbewertung</li> <li>- Pipelineverarbeitung</li> <li>- Speichersysteme</li> <li>- Konzepte der Parallelverarbeitung und parallele Rechnerarchitekturen</li> </ul>
<b>Qualifikationsziele</b>	<p><b>Hardwarepraktikum II:</b>  Im Praktikum wenden die Studierenden Entwurfsmethoden der Digitaltechnik praktisch an. Neben allgemeinen Kompetenzen wie der zeitlichen Ablaufplanung des Praktikums und der sprachlichen Präsentation der Resultate werden manuelle Fertigkeiten beim Schaltungsaufbau sowie die Verknüpfung von technischem und theoretischem Wissen gefördert.</p> <p><b>Rechnerarchitektur:</b>  Die Studierenden sollen strukturelle, organisatorische und implementierungstechnische Aspekte verschiedener Rechnerarchitekturen interpretieren können. Des Weiteren sollen sie in die Lage versetzt werden, die Leistung derartiger Systeme bewerten zu können, wozu sie verschiedene Verfahren und Methoden anwenden. Ein besonderes Augenmerk liegt auf den Möglichkeiten der Parallelarbeit und den damit verbundenen Rechnerarchitekturvarianten, die hinsichtlich ihres Einsatzspektrums sowie der Vor- und Nachteile eingeordnet werden können.</p>
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Verständnis zu allgemeinen Hardwaregrundlagen insbesondere zu digitalen Schaltungen

<b>Literaturhinweise</b>	<b>Hardwarepraktikum II:</b> - Aufgabenspezifische Versuchsanleitungen zu den Praktikumsversuchen  <b>Rechnerarchitektur:</b> - R. Hellmann: „Rechnerarchitektur“, Oldenbourg, in der aktuellen Auflage. - A. Böttcher: „Rechneraufbau und Rechnerarchitektur“, Springer, in der aktuellen Auflage. - A. S. Tanenbaum: „Computerarchitektur“, Pearson Studium, in der aktuellen Auflage. - T. Rauber, G. Rüniger: „Parallele Programmierung“, Springer, in der aktuellen Auflage. - H. G. Kruse: „Leistungsbewertung bei Computersystemen“, Springer, in der aktuellen Auflage.
<b>Aktuelle Lehrressourcen</b>	<b>Hardwarepraktikum II:</b> keine  <b>Rechnerarchitektur:</b> keine
<b>Hinweise</b>	<b>Hardwarepraktikum II:</b> Prüfung: Praktikumsversuche (PX), die jeweils zu mindestens 50% erfolgreich bearbeitet sein müssen
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Datenbanken Database Systems
<b>Modulnummer</b>	C719 [INB3050, MIB3050, MIB-BI3050] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Wintersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr.-Ing. Thomas Kudraß <a href="mailto:thomas.kudrass@htwk-leipzig.de">thomas.kudrass@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr.-Ing. Thomas Kudraß <a href="mailto:thomas.kudrass@htwk-leipzig.de">thomas.kudrass@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	5 ECTS-Punkte
<b>Workload</b>	150 Stunden
<b>Lehrveranstaltungen</b>	4 SWS (2 SWS Vorlesung   2 SWS Seminar)
<b>Selbststudienzeit</b>	94 Stunden
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Projektarbeit
<b>Prüfungsleistung(en)</b>	Prüfung Klausurarbeit Modulprüfung   Prüfungsdauer: 120 Minuten   Wichtigung: 100%   nicht kompensierbar
<b>Lehr- und Lernformen</b>	-
<b>Medienform</b>	keine Angabe
<b>Lehrinhalte/Gliederung</b>	<ul style="list-style-type: none"> <li>- Grundkonzepte von Datenbanken</li> <li>- Entity-Relationship-Modellierung</li> <li>- Relationales Datenmodell (Grundlagen, Relationenalgebra &amp; Relationenkalkül)</li> <li>- Logischer Datenbankentwurf (Modelltransformationen, Normalisierung)</li> <li>- Datenbanksprache SQL: Anfragen, DDL, DML</li> <li>- Integritätssicherung in Datenbanken: Constraints und Trigger</li> <li>- Transaktionen</li> <li>- Datensicherheit und Datenschutz</li> <li>- Erweiterungen relationaler Datenbanksysteme</li> <li>- praktische Übungen mit dem Datenbanksystem Oracle</li> </ul>

<b>Qualifikationsziele</b>	Nach erfolgreichem Abschluss des Moduls verfügt der Student über umfangreiche Erfahrungen bei der Nutzung von Datenbanktechnologie in einer anwendungsorientierten Sichtweise. Er kann die wichtigsten technischen Voraussetzungen beim praktischen Einsatz eines Datenbankmanagementsystems (DBMS) in einem Softwareprojekt beurteilen. Er beherrscht die Formulierung von Datenbankabfragen mittels SQL auf einem vorgegebenen Datenbankschema. Er ist in der Lage, einen Datenbankentwurf durchzuführen, ausgehend von einer Anforderungsanalyse, über die Modellierung bis hin zur Umsetzung in einem konkreten DBMS. Dabei kennt er wichtige Entwurfskriterien und kann diese bei der Modellierung der Datenbank berücksichtigen.
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Keine
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- A. Elmasri, S. Navathe: "Grundlagen von Datenbanksystemen - Ausgabe Grundstudium", Pearson Studium, in der aktuellen Auflage.</li> <li>- A. Kemper, A. Eickler: "Datenbanksysteme", Oldenbourg, in der aktuellen Auflage.</li> <li>- T. Kudraß: "Taschenbuch Datenbanken", Hanser-Verlag, 2007.</li> <li>- K. Ramakrishnan, J. Gehrke: "Database Systems", McGraw-Hill, in der aktuellen Auflage.</li> <li>- Weitere aktuelle Literaturhinweise unter <a href="http://www.kudrass.de">www.kudrass.de</a></li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	Projekt (PVJ): Datenbank-Projekt (2 Belege und Praktikum)
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Technisches Englisch und Schlüsselqualifikationen Technical English and Key Qualifications
<b>Modulnummer</b>	C745 [INB3069, MIB3069, MIB-BI4069] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	2 Semester
<b>Turnus</b>	Sommersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. phil. Uwe Bellmann <a href="mailto:uwe.bellmann@htwk-leipzig.de">uwe.bellmann@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	<p>Prof. Dr. phil. Uwe Bellmann <a href="mailto:uwe.bellmann@htwk-leipzig.de">uwe.bellmann@htwk-leipzig.de</a> Dozent/-in in: "Technisches Englisch"</p> <p>Prof. Dr. rer. nat. Karsten Weicker <a href="mailto:karsten.weicker@htwk-leipzig.de">karsten.weicker@htwk-leipzig.de</a> Dozent/-in in: "Grundlagen wissenschaftlichen Arbeitens"</p> <p>Dr. rer. nat. Martin Schubert <a href="mailto:martin.schubert@htwk-leipzig.de">martin.schubert@htwk-leipzig.de</a> Dozent/-in in: "Studium generale"</p>
<b>Sprache(n)</b>	<p>Englisch in "Technisches Englisch"</p> <p>Deutsch in "Grundlagen wissenschaftlichen Arbeitens"</p> <p>Deutsch in "Studium generale"</p>
<b>ECTS-Leistungspunkte</b>	7 ECTS-Punkte
<b>Workload</b>	<p>194 Stunden</p> <p>120 Stunden in "Technisches Englisch"</p> <p>60 Stunden in "Grundlagen wissenschaftlichen Arbeitens"</p> <p>14 Stunden in "Studium generale"</p>
<b>Lehrveranstaltungen</b>	<p>7 SWS (2 SWS Vorlesung   2 SWS Übung   2 SWS Praktikum   1 SWS Seminar)</p> <p>4 SWS (2 SWS Übung   2 SWS Praktikum) in "Technisches Englisch"</p> <p>2 SWS (1 SWS Vorlesung   1 SWS Seminar) in "Grundlagen wissenschaftlichen Arbeitens"</p> <p>1 SWS (1 SWS Vorlesung) in "Studium generale"</p>



<b>Selbststudienzeit</b>	<b>138 Stunden</b> 60 Stunden E-Learning - Technisches Englisch 10 Stunden Vorbereitung Prüfung - Technisches Englisch 22 Stunden Vorbereitung Lehrveranstaltung - Technisches Englisch 46 Stunden in "Grundlagen wissenschaftlichen Arbeitens" 0 Stunden in "Studium generale"
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Hausarbeit in "Technisches Englisch"  Prüfungsvorleistung Prüfung Computerarbeit in "Technisches Englisch"
<b>Prüfungsleistung(en)</b>	<b>Prüfung Referat</b> Prüfungsdauer: 15 Minuten   Wichtigung: 33.33% in "Technisches Englisch"  <b>Prüfung Computerarbeit</b> Prüfungsdauer: 90 Minuten   Wichtigung: 33.33% in "Technisches Englisch"  <b>Prüfung Projektarbeit</b> Prüfungsdauer: 10 Wochen   Wichtigung: 33.33% in "Grundlagen wissenschaftlichen Arbeitens"  <b>Prüfung Testat</b> Prüfungsdauer: 14 Wochen   Wichtigung: 0%   nicht benotet   nicht kompensierbar in "Studium generale"
<b>Lehr- und Lernformen</b>	<b>Technisches Englisch:</b> - Einzel- und Projektgruppenarbeit - verschiedene PVH und deren Präsentation und Diskussion in der Gruppe  <b>Grundlagen wissenschaftlichen Arbeitens:</b> -  <b>Studium generale:</b> -
<b>Medienform</b>	<b>Technisches Englisch:</b> - Präsentation, mündliche F2F-Interaktion - e-Xplore Technical English, ein WebCourse - Übungsblätter, Handouts - Tafelbild - AV-Trainingsmaterialien von der Festplatte oder aus dem Internet via Datenprojektor  <b>Grundlagen wissenschaftlichen Arbeitens:</b> keine Angabe  <b>Studium generale:</b> keine Angabe

<p><b>Lehrinhalte/Gliederung</b></p>	<p><b>Technisches Englisch:</b></p> <ul style="list-style-type: none"> <li>- General and business English, e.g. presentations and public speaking in English, business contacts face-to-face and on the phone, the language of English lectures, basics of traditional commercial and email correspondence including job applications, CVs, and covering letters</li> <li>- English for specific purposes</li> <li>• Terminology</li> <li>• Basics and current trends in computer science</li> <li>• Technical English for students of science and engineering, e.g. numbers, mathematical symbols and operations, databases, complex systems, programming, spreadsheets, product lifecycle management, electronic learning, licenses etc.</li> <li>- Grammar, e.g. adjectives, adverbs, articles, prepositions, pronouns, sentences, verbs, cohesion, word formation etc.</li> </ul> <p><b>Grundlagen wissenschaftlichen Arbeitens:</b></p> <ul style="list-style-type: none"> <li>- Themen: Literaturrecherche, Informatik als Wissenschaft, wissenschaftlich Schreiben, Einführung in Latex, Begutachtung wissenschaftlicher Arbeiten, Wissenschaftsethik, wissenschaftliche Vorträge</li> <li>- Erarbeitung, gegenseitige Begutachtung und Präsentation einer eigenen Arbeit entsprechend der typischen Organisation einer wissenschaftlichen Tagung</li> </ul> <p><b>Studium generale:</b></p> <p>Im Studium generale werden gesellschaftsrelevante Themen und wissenschaftlich/technologische Fragestellungen mit fachübergreifendem Charakter behandelt. Dabei soll der Blick auf die Funktions- und Kommunikationsmechanismen in unserer Gesellschaft geschärft werden. Die Bearbeitung eines Themas erfolgt aus möglichst unterschiedlichen Perspektiven.</p> <p>Zur Realisierung des Lernziels werden Lehrveranstaltungen mit unterschiedlichen Lehrinhalten angeboten, aus denen je nach Platzangebot frei gewählt werden kann.</p>
--------------------------------------	--

<b>Qualifikationsziele</b>	<p>Durch das Training ausgewählter Schlüsselqualifikationen, werden die Studierenden dazu befähigt, als Informatiker im beruflichen Anwendungskontext zu arbeiten. Hierzu zählt die erfolgreiche Auseinandersetzung mit englischsprachiger Fachliteratur, technisches/wissenschaftliches Schreiben, das Halten einer Präsentation sowie die Fähigkeit, über das eigene Fachgebiet hinauszudenken.</p> <p><b>Technisches Englisch:</b></p> <p>Die Studierenden besitzen anwendungsbereite Kenntnisse und Fähigkeiten in Englisch für die fach- und berufsbezogene Kommunikation auf Niveau Mittelstufe bis Oberstufe.</p> <p>Erfolgreiche Teilnehmer können die englische Sprache in beruflichen Situationen und Kontexten (Informatik, Wirtschaft und IT) erfolgreich verwenden, z. B. Fachtexte flüssig lesen, Fachvorträge verstehen und in Gesprächen und Vorträgen eigene Standpunkte vertreten.</p> <p><b>Grundlagen wissenschaftlichen Arbeitens:</b></p> <p>Die Studierenden können zu einem vorgegebenen Thema der Informatik selbständig Literatur suchen und bewerten, ihre eigene Arbeit in die Literatur einbetten, wissenschaftliche oder technische Arbeiten von anderen begutachten, eine technische/wissenschaftliche Abhandlung unter Berücksichtigung typischer Konventionen des Fachgebiets schreiben und eine Beamer-gestützte Präsentation halten</p> <p><b>Studium generale:</b></p> <p>Im Studium generale sollen der fachübergreifende Charakter von Lehre und Forschung sowie die Zusammenhänge von Theorie und Praxis vermittelt werden. Die Studierenden sollen dabei befähigt werden, über ihr eigenes Handeln zu reflektieren, ihr Wissen einzuordnen und Zusammenhänge zu erkennen. Durch die offene und kontroverse Auseinandersetzung anhand eines ausgewählten Themas soll das Urteils- und Handlungsvermögen in politischen, ökonomischen, ökologischen und interkulturellen Bereichen ausgebildet werden.</p>
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	<p><b>Technisches Englisch:</b></p> <p>Fachhochschulreife mit Englischkenntnissen auf mittlerem Niveau.</p> <p>Bei Bedarf sollte zur Auffrischung der Vorkenntnisse zusätzlich ein Refresher-Course belegt werden.</p> <p><b>Grundlagen wissenschaftlichen Arbeitens und Studium generale:</b></p> keine
<b>Literaturhinweise</b>	<p><b>Technisches Englisch:</b></p> <ul style="list-style-type: none"> <li>- Weitere aktuelle Literaturhinweise werden in den Lehrveranstaltungen gegeben.</li> </ul> <p><b>Grundlagen wissenschaftlichen Arbeitens:</b></p> <ul style="list-style-type: none"> <li>- H. Balzert et al.: "Wissenschaftliches Arbeiten - Wissenschaft, Quellen, Artefakte, Organisation,</li> <li>- Präsentation" W3L, in der aktuellen Auflage.</li> </ul> <p><b>Studium generale:</b></p> <ul style="list-style-type: none"> <li>- themenspezifisch</li> </ul>

<b>Aktuelle Lehrressourcen</b>	<b>Technisches Englisch:</b> www.webcourses.de  <b>Grundlagen wissenschaftlichen Arbeitens:</b> keine  <b>Studium generale:</b> keine
<b>Hinweise</b>	<b>Technisches Englisch:</b> Prüfungsvorleistung: PVH und PVC (erfolgreicher Abschluss des WebCourses) - WebCourse beinhaltet Workload von 60 Stunden und ist mit 2 SWS ausgewiesen.
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	<a href="https://webcourses.htwk-leipzig.de/xte">https://webcourses.htwk-leipzig.de/xte</a>

<b>Modul</b>	Softwaretechnik Software Engineering
<b>Modulnummer</b>	C559 [INB3070, MIB3070, MIB-BI3070] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Wintersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. Karsten Weicker <a href="mailto:karsten.weicker@htwk-leipzig.de">karsten.weicker@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. Karsten Weicker <a href="mailto:karsten.weicker@htwk-leipzig.de">karsten.weicker@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	5 ECTS-Punkte
<b>Workload</b>	150 Stunden
<b>Lehrveranstaltungen</b>	4 SWS (2 SWS Vorlesung   2 SWS Seminar)
<b>Selbststudienzeit</b>	94 Stunden 30 Stunden Sonstiges 64 Stunden Vorbereitung Lehrveranstaltung
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Projektarbeit  Prüfungsvorleistung Testat
<b>Prüfungsleistung(en)</b>	Prüfung Klausurarbeit Prüfungsdauer: 120 Minuten   Wichtigung: 100%
<b>Lehr- und Lernformen</b>	-
<b>Medienform</b>	keine Angabe

<b>Lehrinhalte/Gliederung</b>	<ul style="list-style-type: none"> <li>- Überblick über den Software-Lebenszyklus, Gesetzmäßigkeiten des Software Engineering</li> <li>- Anforderungsspezifikation (UML, GUI-Prototypen)</li> <li>- Entwurf (Architekturprinzipien, Überblick über Software-Architekturen, Grob- und Feinentwurf, Entwurfsmuster)</li> <li>- Implementierung (Programmierrichtlinien, Unit-Tests, Refactoring, Versionsmanagement)</li> <li>- Projektmanagement (agile Software-Entwicklung, Prozessmodelle, Kostenschätzung, Aspekte der Planung, Reengineering-Projekte)</li> </ul>
<b>Qualifikationsziele</b>	<p>Die Studierenden können Dokumente aus den unterschiedlichen Phasen der Softwareentwicklung lesen, für kleine Projekte selbst erstellen und kritisch hinsichtlich der Qualität bewerten. Sie beherrschen Notationen und Werkzeuge der UML-Modellierung und der Anforderungsspezifikation.</p> <p>Ferner können sie existierende Projekte hinsichtlich der Software-Architektur untersucht sowie für kleine Projekte selbige entwickeln und umsetzen. Werkzeuge zum Testen von Software, Refactoring, Versionsmanagement und Quelltextdokumentation werden beherrscht</p>
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Programmierkompetenzen sollten soweit vorhanden sein, dass kleine Programme mit graphischer Benutzeroberfläche erstellt werden können.
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- J. Ludewig, H. Lichter: "Software Engineering", dpunkt, in der aktuellen Auflage.</li> <li>- A. Endres, D. Rombach: "A Handbook of Software and Systems Engineering", Pearson, 2003.</li> <li>- C. Rupp et al.: "UML 2 glasklar. Praxiswissen für die UML-Modellierung", Hanser, in der aktuellen Auflage.</li> <li>- G. Starke: "Effektive Software-Architekturen", Hanser, in der aktuellen Auflage.</li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	<p>Prüfungsvorleistungen:</p> <p>Testat (PVT): wöchentliche Bearbeitung von Aufgaben im Seminar  Projekt (PVJ): erfolgreiche Bearbeitung eines Anwendungsprojekts in kleinen Teams</p>
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Fortgeschrittene Programmierung Advanced Programming
<b>Modulnummer</b>	C393 [INB4010, MIB4010, MIB-BI4010] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Sommersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. Johannes Waldmann <a href="mailto:johannes.waldmann@htwk-leipzig.de">johannes.waldmann@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. Johannes Waldmann <a href="mailto:johannes.waldmann@htwk-leipzig.de">johannes.waldmann@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	5 ECTS-Punkte
<b>Workload</b>	150 Stunden
<b>Lehrveranstaltungen</b>	4 SWS (2 SWS Vorlesung   2 SWS Seminar)
<b>Selbststudienzeit</b>	94 Stunden
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Beleg
<b>Prüfungsleistung(en)</b>	Prüfung Klausurarbeit Modulprüfung   Prüfungsdauer: 120 Minuten   Wichtigung: 100%   nicht kompensierbar
<b>Lehr- und Lernformen</b>	-
<b>Medienform</b>	keine Angabe
<b>Lehrinhalte/Gliederung</b>	<ul style="list-style-type: none"> <li>- algebraische Datentypen, Pattern Matching, Termersetzung</li> <li>- Funktionen (polymorph getypt, von höherer Ordnung), Lambda-Kalkül, Rekursionsmuster (map, fold)</li> <li>- Typklassen, Interfaces, Unit-Tests, automatische Testfallerzeugung</li> <li>- Entwurfsmuster für Programme mit Zustandsänderungen</li> <li>- Bedarfsauswertung, unendliche Datenstrukturen, Iteratoren</li> <li>- Codequalität, Code smells, Refaktorisierung</li> </ul>

<b>Qualifikationsziele</b>	Die Studierenden haben fortgeschrittene Konzepte der Programmierung sowie ihre Ausprägungen in verschiedenen Programmiersprachen erlernt. Sie können diese Konzepte bei konkreten Programmieraufgaben anwenden.
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Keine
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- M. Naftalin, P. Wadler: "Java generics and Collections", O'Reilly, 2006.</li> <li>- B. O'Sullivan, D. Stewart, J. Goerzen: "Real World Haskell", O'Reilly, 2008.</li> <li>- E. Gamma, R. Helm, R. E. Johnson: "Design Patterns", Addison-Wesley, 1995.</li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	<p>Prüfungsvorleistung:</p> <p>Belege (PVB): Regelmäßiges und erfolgreiches Bearbeiten von Übungsaufgaben</p>
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	



<b>Modul</b>	Softwareprojekt Software Engineering Project
<b>Modulnummer</b>	C732 [INB4080, MIB4080, MIB-BI4080] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	2 Semester
<b>Turnus</b>	Wintersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. Karsten Weicker <a href="mailto:karsten.weicker@htwk-leipzig.de">karsten.weicker@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. Karsten Weicker <a href="mailto:karsten.weicker@htwk-leipzig.de">karsten.weicker@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch in "Softwareprojekt Teil 1"  Deutsch in "Softwareprojekt Teil 2"
<b>ECTS-Leistungspunkte</b>	8 ECTS-Punkte
<b>Workload</b>	240 Stunden 90 Stunden in "Softwareprojekt Teil 1" 150 Stunden in "Softwareprojekt Teil 2"
<b>Lehrveranstaltungen</b>	2 SWS (1 SWS Vorlesung   1 SWS Praktikum) 1 SWS (0.50 SWS Vorlesung   0.50 SWS Praktikum) in "Softwareprojekt Teil 1" 1 SWS (0.50 SWS Vorlesung   0.50 SWS Praktikum) in "Softwareprojekt Teil 2"
<b>Selbststudienzeit</b>	180 Stunden 76 Stunden Bearbeitung Prüfungsleistung - Softwareprojekt Teil 1 136 Stunden Bearbeitung Prüfungsleistung - Softwareprojekt Teil 2
<b>Prüfungsvorleistung(en)</b>	Keine
<b>Prüfungsleistung(en)</b>	Prüfung Projektarbeit Modulprüfung   Prüfungsdauer: 9 Monate   Wichtigung: 100%
<b>Lehr- und Lernformen</b>	<b>Softwareprojekt Teil 1:</b> -  <b>Softwareprojekt Teil 2:</b> keine Angabe

<b>Medienform</b>	<b>Softwareprojekt Teil 1:</b> -  <b>Softwareprojekt Teil 2:</b> keine Angabe
<b>Lehrinhalte/Gliederung</b>	<b>Softwareprojekt Teil 1:</b> - Vorstellung der Anforderungen - Teambildung - Erstellung einer Anforderungsspezifikation und einer Architekturvision mit Präsentationen an mehreren Meilensteinen - Erstellung einer produktiv einsetzbaren Software mit Präsentationen an mehreren Meilensteinen  <b>Softwareprojekt Teil 2:</b> - Erstellung einer Anforderungsspezifikation und einer Architekturvision mit Präsentationen an mehreren Meilensteinen - Erstellung einer produktiv einsetzbaren Software mit Präsentationen an mehreren Meilensteinen - Poster-Abschlusspräsentation nach der ersten Phase - Wartungsphase, in der Fehler behoben und neue Anforderungen umgesetzt werden - Abschlusspräsentation als Vortrag
<b>Qualifikationsziele</b>	Die Studierenden können sich an allen Phasen eines großen Softwareprojekts im Rahmen eines vorgegebenen agilen Vorgangmodells (Scrum) beteiligen. Hierzu gehören insbesondere die folgenden Kompetenzen. Arbeitspakete können im Detail selbständig geplant, termingerecht bearbeitet und dokumentiert werden. Sie können mit einem Dokumenten-Repository zum Versionsmanagement umgehen. Sie können fremden Quelltext lesen, darin Entwurfskonzepte erkennen sowie Änderungen durchführen. Sie erkennen selbständig Schnittstellen zu den Arbeitspaketen anderer Teammitglieder, können die Probleme benennen und selbständig Absprachen durchführen. Sie können für die konkreten Anforderungen einer zu erstellenden Anwendung Artefakte der Software-Entwicklung erstellen bzw. substantiell dazu beitragen. Insbesondere sind sie in der Lage Teilmodule zu entwerfen und im Rahmen der Gesamtsoftware umzusetzen. Innerhalb des Projektkontexts beherrschen sie erfolgreich Strategien zur Qualitätssicherung, d.h. Fehlermanagement, Uni-Tests und Reviews. Die Qualität von Artefakten kann im Rahmen von Reviews beurteilt werden. Darüber hinaus werden im Projektkontext Probleme hinsichtlich der Planung und Durchführbarkeit erkannt sowie Maßnahmen vorgeschlagen. Die Studierenden erkennen Konflikte im Team und können Strategien zur Konfliktlösung anwenden. Selbstkompetenzen, wie Verbindlichkeit, Disziplin, Termintreue, Kompromissbereitschaft und die Übernahme von Verantwortung, werden projektdienlich entwickelt und eingesetzt.
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Programmierkompetenzen sollten soweit vorhanden sein, dass kleine Programme mit graphischer Benutzeroberfläche erstellt werden können

<b>Literaturhinweise</b>	<b>Softwareprojekt Teil 1:</b> - J. Ludewig, H. Lichte: "Software Engineering", dpunkt, in der aktuellen Auflage. - C. Rupp et al.: "UML 2 glasklar. Praxiswissen für die UML-Modellierung", Hanser, in der aktuellen Auflage. - H. Kellner: "Soziale Kompetenz für Ingenieure, Informatiker und Naturwissenschaftler", Hanser, 2006. - U. Vogenschow, B. Schneider: "Soft Skills für Softwareentwickler", dpunkt, in der aktuellen Auflage. - R. Pichler: "Scrum - Agiles Projektmanagement erfolgreich einsetzen", dpunkt, 2007.  <b>Softwareprojekt Teil 2:</b> siehe Teil 1
<b>Aktuelle Lehrressourcen</b>	<b>Softwareprojekt Teil 2:</b> keine
<b>Hinweise</b>	Keine Angabe
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	IT-Sicherheit IT Security
<b>Modulnummer</b>	C799 [INB5010, MIB5010, MIB-BI5010] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Wintersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. Uwe Petermann <a href="mailto:uwe.petermann@htwk-leipzig.de">uwe.petermann@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. Uwe Petermann <a href="mailto:uwe.petermann@htwk-leipzig.de">uwe.petermann@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	5 ECTS-Punkte
<b>Workload</b>	150 Stunden
<b>Lehrveranstaltungen</b>	4 SWS (2 SWS Vorlesung   2 SWS Seminar)
<b>Selbststudienzeit</b>	94 Stunden
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Präsentation
<b>Prüfungsleistung(en)</b>	Prüfung Klausurarbeit Modulprüfung   Prüfungsdauer: 90 Minuten   Wichtung: 100%   nicht kompensierbar
<b>Lehr- und Lernformen</b>	-
<b>Medienform</b>	keine Angabe
<b>Lehrinhalte/Gliederung</b>	<ul style="list-style-type: none"> <li>- Methode nach IT-Grundschutz zur systematischen Entwicklung von Sicherheitskonzepten.</li> <li>- Security Management nach ITIL (IT Infrastructure Library)</li> <li>- Umsetzung von Sicherheitskonzepten mit Mitteln der Hard- und Software</li> <li>- Grundlegende Kenntnisse zu rechtlichen Belangen der IT-Sicherheit</li> <li>- Praktische Übungen zur Realisierung von Maßnahmen der Sicherheit</li> </ul>

<b>Qualifikationsziele</b>	Nach erfolgreichem Abschluss der Lehrveranstaltung sind die Studierenden in der Lage, Bedrohungen von Rechnern und Netzen zu erkennen und den Schutzbedarf dieser Ressourcen einzuschätzen. Sie sind mit der Systematik der Zertifizierung der IT-Sicherheit von Organisationen nach internationalen Normen wie ISO 27001 vertraut und können in Organisationen, die sich einer Zertifizierung unterziehen, als Ansprechpartner der Auditoren wirken.
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Die Studierenden sind sowohl mit den Wirkprinzipien von Rechnern, der Rolle und Funktionsweise von Betriebssystemen sowie mit der Kommunikation von Rechnern über Netze vertraut.
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- R. J. Anderson: "Security Engineering", Wiley, 2010.</li> <li>- C. Eckert. : "IT-Sicherheit", Oldenburg, 2008.</li> <li>- H. Kersten et al.: "IT-Sicherheitsmanagement nach ISO 27001 und Grundschutz", Vieweg, 2008 .</li> <li>- K. Mitnik, W. Simon: "Die Kunst der Täuschung", mitp, 2011.</li> <li>- A. Olbrich: "ITIL kompakt und verständlich", Vieweg, 2006.</li> <li>- M. Schumacher et al.: "Hacker Contest", Springer, 2003.</li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	Keine Angabe
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Einführung in die Betriebswirtschaftslehre Introduction to Business Administration
<b>Modulnummer</b>	W861 [INB5060, MIB8130, MIB-BI8460] Version: 1
<b>Fakultät</b>	FWW: Fakultät Wirtschaftswissenschaft und Wirtschaftsingenieurwesen
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Wintersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. oec. publ. Sabine Hüttinger <a href="mailto:sabine.huettinger@htwk-leipzig.de">sabine.huettinger@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Dipl.-Kaufrau Gisela Schwetzler <a href="mailto:gisela.schwetzler@htwk-leipzig.de">gisela.schwetzler@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	5 ECTS-Punkte
<b>Workload</b>	150 Stunden
<b>Lehrveranstaltungen</b>	4 SWS (2 SWS Vorlesung   2 SWS Seminar)
<b>Selbststudienzeit</b>	94 Stunden 30 Stunden Bearbeitung Prüfungsvorleistung 64 Stunden Vorbereitung Lehrveranstaltung
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Referat
<b>Prüfungsleistung(en)</b>	Prüfung Klausurarbeit Modulprüfung   Prüfungsdauer: 90 Minuten   Wichtung: 100%
<b>Lehr- und Lernformen</b>	keine Angabe
<b>Medienform</b>	keine Angabe

<b>Lehrinhalte/Gliederung</b>	<ul style="list-style-type: none"> <li>- Unternehmen und Umwelt</li> <li>- Typologie</li> <li>- Rechnungswesen intern (Kostenrechnung) und extern (Jahresabschluss)</li> <li>- Existenzgründung mit Businessplan</li> <li>- Marketing</li> <li>- Steuern</li> <li>- Insolvenzverfahren</li> <li>- Investitionsrechnung</li> <li>- Finanzierung</li> <li>- Controlling</li> <li>- Führung</li> </ul>
<b>Qualifikationsziele</b>	<p>Ziel ist die Vermittlung von grundlegenden betriebswirtschaftlichen Kenntnissen und Fertigkeiten.</p> <p>Fach- und methodische Kompetenzen:          Kennen betriebswirtschaftlicher Begriffe und Denkweisen, Verstehen wichtiger betriebswirtschaftlicher Zusammenhänge, kunden- und kostenorientiertes Denken am Arbeitsplatz. Die Einführung in die Betriebswirtschaftslehre ermöglicht den Informatikern eine interdisziplinäre Sicht, die sie in ihrer beruflichen Entwicklung auch im Hinblick auf Führungsaufgaben unterstützen wird.</p>
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Keine
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- J. Drukarczyk: „Finanzierung“, UTB, in der aktuellen Auflage.</li> <li>- H. Meffert: „Marketing“, Gabler, in der aktuellen Auflage.</li> <li>- J. Thommen, A. Achleitner: „Allgemeine Betriebswirtschaftslehre“, Gabler, in der aktuellen Auflage.</li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine Angabe
<b>Hinweise</b>	<p>Prüfungsvorleistung:</p> <p>Referat (PVR): Referat mit max. 4 Teilnehmern</p>
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Praxisprojekt Practical Project
<b>Modulnummer</b>	C222 [INB6000, MIB6000, MIB-BI6000] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Sommersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr.-Ing. Thomas Kudraß <a href="mailto:thomas.kudrass@htwk-leipzig.de">thomas.kudrass@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr.-Ing. Thomas Kudraß <a href="mailto:thomas.kudrass@htwk-leipzig.de">thomas.kudrass@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	15 ECTS-Punkte
<b>Workload</b>	450 Stunden
<b>Lehrveranstaltungen</b>	0 SWS
<b>Selbststudienzeit</b>	0 Stunden
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Beleg
<b>Prüfungsleistung(en)</b>	Prüfung Präsentation Prüfungsdauer: 30 Minuten   Wichtigung: 100%
<b>Lehr- und Lernformen</b>	-
<b>Medienform</b>	Keine Angabe
<b>Lehrinhalte/Gliederung</b>	- themenspezifisch



<b>Qualifikationsziele</b>	<p>Ziele: Das Praxisprojekt wird in einem Unternehmen oder in einer anderen Einrichtung der Berufspraxis abgeleistet. Es dient der Vermittlung praktischer Erfahrungen und Fähigkeiten zur Ergänzung der theoretischen Kenntnisse.</p> <p>Kompetenzen: Der Studierende soll den Einsatz seiner Fachkenntnisse in der Praxis üben, praktische Aufgaben und Zusammenhänge abstrahieren lernen und seine Kommunikations- und Teamfähigkeit ausbauen. Abschließend soll er seine Fähigkeit unter Beweis stellen, die eigene Tätigkeit im Praxisprojekt kompakt im Rahmen eines Vortrages oder eines Posters darzustellen.</p> <p>Einbindung in die Berufsvorbereitung: Das Praxisprojekt dient der unmittelbaren Berufsvorbereitung. Es kann sehr gut zu einer persönlichen Sondierung und Kontaktherstellung zu potenziellen späteren Arbeitgebern genutzt werden.</p>
<b>Zulassungsvoraussetzung</b>	Festlegung durch Prüfungsordnung und Praktikumsordnung
<b>Empfohlene Voraussetzungen</b>	Keine
<b>Literaturhinweise</b>	- themenspezifisch
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	<p>Workload: 450h entsprechen 12 Wochen Tätigkeit auf einer Praxisstelle</p> <p>Prüfungsvorleistung Beleg (PVB): Praktikumsbericht des Studenten Tätigkeitsnachweis der Praxisstelle</p>
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Grundlagen der mobilen Robotik Principles of mobile robotics
<b>Modulnummer</b>	C010 Version: 0
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Wintersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. Jens Wagner <a href="mailto:jens.wagner@htwk-leipzig.de">jens.wagner@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. Jens Wagner <a href="mailto:jens.wagner@htwk-leipzig.de">jens.wagner@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	5 ECTS-Punkte
<b>Workload</b>	150 Stunden
<b>Lehrveranstaltungen</b>	4 SWS (2 SWS Vorlesung   2 SWS Übung)
<b>Selbststudienzeit</b>	94 Stunden 60 Stunden Bearbeitung Prüfungsleistung 34 Stunden Bearbeitung Prüfungsvorleistung
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Laborarbeit
<b>Prüfungsleistung(en)</b>	Prüfung Hausarbeit Prüfungsdauer: 3 Monate   Wichtung: 50%   nicht kompensierbar  Prüfung Referat Prüfungsdauer: 20 Minuten   Wichtung: 50%   nicht kompensierbar
<b>Lehr- und Lernformen</b>	keine Angabe
<b>Medienform</b>	keine Angabe

<b>Lehrinhalte/Gliederung</b>	<ul style="list-style-type: none"> <li>- Begriffe, Beispielimplementierungen</li> <li>- Einstieg in das wissenschaftliche Arbeiten</li> <li>- Sensorik</li> <li>- Aktoren</li> <li>- Navigation</li> <li>- Roboterkontrollarchitekturen</li> <li>- Designbeispiele: Fußballroboter, Lernroboter, Staubsaugroboter</li> <li>- Projekt und Präsentation</li> </ul>
<b>Qualifikationsziele</b>	<p>Die Studierenden sind in der Lage mobile Roboter zu klassifizieren und an Hand ihres Aufbaus zu analysieren. Die Studierenden verfügen über grundsätzliches Wissen zum Aufbau mobiler Roboter und können komplexe praktische Aufgaben im Labor planen und lösen. Einzelne Aspekte werden übergewichtet, um sie im Detail zu behandeln: z.B. Ultraschallsensoren, Kartographie, Wegfindung, Selbstortung. Die Studierenden arbeiten mit aktueller wissenschaftlicher Literatur, einschließlich Monographien und sind in der Lage Ihre eigenen Ergebnisse zu verschriftlichen.</p>
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	<p>Theoretische und praktische Erfahrungen im Algorithmieren, Programmieren sowie Arbeiten mit Datenstrukturen in einem systemnahen Softwareentwicklungssystem, Beherrschen von physikalischen und logischen Grundlagen der Digitaltechnik, deren Entwurfsmethoden sowie der digitalen Mess- und Analysewerkzeuge. Praktische Erfahrung mit einem einfachen Mikrocontrollerentwicklungssystem.</p>
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- Joachim Hertzberg et al: „Mobile Roboter“, Springer 2012</li> <li>- Ulrich Nehmzow: „Mobile Robotik“, Springer 2013</li> <li>- André Araújo et al. „Integrating Arduino-Based Educational Mobile Robots in ROS“, Journal of Intelligent &amp; Robotic Systems, Volume 77, Issue2, 2015</li> <li>- Saskia Uta Dübener et al.: „Gegenüberstellung von kostengünstigen Robotern als Lernobjekte ...“, Skill – Studierendenkonferenz, Chemnitz, 2017</li> <li>- Anina Ambra Morgner: <a href="https://www.youtube.com/watch?v=ZpcrRbYR64k">https://www.youtube.com/watch?v=ZpcrRbYR64k</a>, 2017</li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	<p>Prüfungsprojekt: Individuelle Aufgabenstellungen auf einer einheitlichen, vorgegebenen Plattform. Die Plattform unterliegt jährlichen Verbesserungen, daher ändern sich die Themen regelmäßig.</p>
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Dokumentbeschreibungssprachen Document Description Languages
<b>Modulnummer</b>	C651 [MIB-BI5040, INB8040, MIB8040] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Wintersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. habil. Michael Frank <a href="mailto:michael.frank@htwk-leipzig.de">michael.frank@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. habil. Michael Frank <a href="mailto:michael.frank@htwk-leipzig.de">michael.frank@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	5 ECTS-Punkte
<b>Workload</b>	150 Stunden
<b>Lehrveranstaltungen</b>	4 SWS (2 SWS Vorlesung   2 SWS Seminar)
<b>Selbststudienzeit</b>	94 Stunden 34 Stunden Selbststudium 60 Stunden Bearbeitung Prüfungsleistung
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Beleg
<b>Prüfungsleistung(en)</b>	Prüfung Projektarbeit Modulprüfung   Prüfungsdauer: 6 Wochen   Wichtigung: 100%
<b>Lehr- und Lernformen</b>	-
<b>Medienform</b>	-

<b>Lehrinhalte/Gliederung</b>	<ul style="list-style-type: none"> <li>- Einführung in XML als Datentransport und als Applikationssteuerungssprache, Einführung in XML-Editoren</li> <li>- Wohlgeformtheit und Gültigkeit von Dokumenten</li> <li>- Strukturdefinition mit Document Type Definition (DTD)</li> <li>- Darstellung von XML-Inhalten als Webseiten mit CSS</li> <li>- Darstellung von XML-Inhalten als textbasierte, über Browser darstellbare Dateien mit XSLT</li> <li>- XML-Schema-Definitionen und ihre verschiedenen Designs</li> <li>- Kurzeinführung in LaTeX</li> <li>- Praktische Übungen aller Aspekte, großes Projekt zum Datentransport und zur Datendarstellung</li> </ul>
<b>Qualifikationsziele</b>	<p>Syntax und Semantik der eXtensible Markup Language (XML), ihrer Strukturdefinitionen Document Type Definition (DTD) und XML-Schema Definition (XSD) und der Darstellungssprache eXtensible Stylesheet Language (XSLT-Fall) werden beherrscht. Anhand eines umfangreichen Programmierprojekts wurden praktische Erfahrungen mit XML-Projekten erworben. Im Umgang mit LaTeX als einer möglichen Umsetzungsform großer Dokumente sind für die Bachelorarbeit anwendbare Fertigkeiten entstanden.</p>
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Beherrschung statischer Webprogrammierung mit HTML und CSS
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- H. Erlenkötter: "XML - Extensible Markup Language von Anfang an", Rowohlt, 2003.</li> <li>- T. Hauser: "XML-Standards. schnell+kompakt.", Entwickler.Press, 2010.</li> <li>- D. Koch: "XSLT schnell+kompakt", Entwickler.Press, 2007.</li> <li>- D. Koch: "XML für Webentwickler. Ein praktischer Einstieg.", Hanser, 2010.</li> <li>- C. Wenz: "Reguläre Ausdrücke schnell+kompakt", Entwickler.Press, 2006.</li> <li>- T. J. Sebestyen: "XML: Einstieg für Anspruchsvolle", Addison-Wesley, 2010.</li> <li>- Spezifikationen des W3C zu den XML-Standards, weitere Empfehlungen im Kurs.</li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	Prüfungsvorleistung Belege (PVB): Übungsfragen und -aufgaben (wöchentlich)
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Hardware-Entwurfstechnik Hardware Design Tools
<b>Modulnummer</b>	C448 [INB8100] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Sommersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. Heinrich Krämer <a href="mailto:heinrich.kraemer@htwk-leipzig.de">heinrich.kraemer@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. Heinrich Krämer <a href="mailto:heinrich.kraemer@htwk-leipzig.de">heinrich.kraemer@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	5 ECTS-Punkte
<b>Workload</b>	150 Stunden
<b>Lehrveranstaltungen</b>	4 SWS (2 SWS Vorlesung   2 SWS Seminar)
<b>Selbststudienzeit</b>	94 Stunden 60 Stunden Bearbeitung Prüfungsvorleistung 34 Stunden Vorbereitung Lehrveranstaltung
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Projekt
<b>Prüfungsleistung(en)</b>	Prüfung mündliches Fachgespräch Prüfungsdauer: 30 Minuten   Wichtigung: 100%
<b>Lehr- und Lernformen</b>	keine Angabe
<b>Medienform</b>	keine Angabe

<b>Lehrinhalte/Gliederung</b>	<ul style="list-style-type: none"> <li>- Manuell entworfene Komponenten <ul style="list-style-type: none"> <li>• Addierer</li> <li>• Multiplizierer</li> <li>• Dividierer</li> </ul> </li> <li>- Logiksynthese <ul style="list-style-type: none"> <li>• Zweistufige Logikminimierung</li> <li>• Mehrstufige Logiksynthese</li> </ul> </li> <li>- Entwurf von Steuerwerken <ul style="list-style-type: none"> <li>• Architekturen von Automaten</li> <li>• Zustandskodierung</li> </ul> </li> <li>- Einführung in VHDL</li> </ul>
<b>Qualifikationsziele</b>	Die Studenten können verschiedene Entwurfsansätze auf der RT-, Logikebene sowie die Arbeitsweise der Entwurfssysteme nachvollziehen. Sie können zu einem gegebenen Problem eine Hardware-Lösung spezifizieren und (insbesondere mit FPGAs) realisieren.
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Entwurf digitaler Schaltungen
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- M. Ercegovac, T. Lang: „Digital Arithmetic“, Morgan Kaufmann Publishers, 2003.</li> <li>- M. Lu: „Arithmetic and Logic in Computersystems“, Wiley, 2004.</li> <li>- J. Reichardt, B. Schwarz: „VHDL-Synthese: Entwurf digitaler Schaltungen und Systeme“, Oldenbourg, 2012.</li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	Keine Angabe
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Numerische Mathematik Numerical Mathematics
<b>Modulnummer</b>	N341 [INB8120] Version: 1
<b>Fakultät</b>	MNZ(Ma): Mathematik - Mathematisch-Naturwissenschaftliches Zentrum
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Wintersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. habil. Hans-Jürgen Dobner <a href="mailto:hans-juergen.dobner@htwk-leipzig.de">hans-juergen.dobner@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. habil. Hans-Jürgen Dobner <a href="mailto:hans-juergen.dobner@htwk-leipzig.de">hans-juergen.dobner@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	5 ECTS-Punkte
<b>Workload</b>	150 Stunden
<b>Lehrveranstaltungen</b>	4 SWS (2 SWS Vorlesung   2 SWS Seminar)
<b>Selbststudienzeit</b>	94 Stunden
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Beleg
<b>Prüfungsleistung(en)</b>	Prüfung Klausurarbeit Prüfungsdauer: 120 Minuten   Wichtigung: 100%
<b>Lehr- und Lernformen</b>	keine Angabe
<b>Medienform</b>	keine Angabe
<b>Lehrinhalte/Gliederung</b>	<ul style="list-style-type: none"> <li>- Grundlagen des num. Rechnens und der Fehleranalyse</li> <li>- Normen von Vektoren und Matrizen</li> <li>- Direkte und iterative Verfahren zur Lösung linearer Gleichungssysteme</li> <li>- Iterationsverfahren zur numerischen Lösung nichtlinearer Gleichungen und Gleichungssysteme</li> <li>- Eigenwertprobleme</li> <li>- Numerische Integration und Differenziation</li> <li>- Numerische Methoden für Anfangswertprobleme gewöhnlicher Differenzialgleichungen.</li> </ul>



<b>Qualifikationsziele</b>	Die Studenten erwerben Kenntnisse bezüglich des Rechnens mit endlicher Stellenzahl und der Fortpflanzung von Fehlern und können diese bei der Analyse einfacher numerischer Verfahren anwenden. Sie lernen grundlegende Verfahren der numerischen Linearen Algebra kennen und erstellen dazu eigene Programme. Bezüglich nichtlinearer Probleme lernen Sie die Iteration als wesentliches Prinzip zu verstehen und die Konvergenzbedingungen zu überprüfen. Durch Programmierung und Test ausgewählter Verfahren erwerben sie Fähigkeiten und Fertigkeiten zur eigenständigen Erstellung numerischer Software.
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Kenntnisse in Analysis und Lineare Algebra
<b>Literaturhinweise</b>	- H. R. Schwarz, N. Köckler: „Numerische Mathematik“, Springer, 2011. - R. Plato: „Numerische Mathematik kompakt“, ViewegTeubner, 2009.
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	Keine Angabe
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Einführung in ERP-Software (SAP) Introduction to ERP Software (SAP)
<b>Modulnummer</b>	N450 [INB8130] Version: 1
<b>Fakultät</b>	MNZ(Ma): Mathematik - Mathematisch-Naturwissenschaftliches Zentrum
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Sommersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. Tobias Martin <a href="mailto:tobias.martin@htwk-leipzig.de">tobias.martin@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. Tobias Martin <a href="mailto:tobias.martin@htwk-leipzig.de">tobias.martin@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	5 ECTS-Punkte
<b>Workload</b>	150 Stunden
<b>Lehrveranstaltungen</b>	4 SWS (2 SWS Vorlesung   2 SWS Übung)
<b>Selbststudienzeit</b>	94 Stunden 30 Stunden Selbststudium 30 Stunden Vorbereitung Prüfung 34 Stunden Vorbereitung Lehrveranstaltung
<b>Prüfungsvorleistung(en)</b>	Keine
<b>Prüfungsleistung(en)</b>	Prüfung Computerarbeit Prüfungsdauer: 90 Minuten   Wichtung: 100%
<b>Lehr- und Lernformen</b>	Vorlesung mit integrierten Übungen
<b>Medienform</b>	keine Angabe

<b>Lehrinhalte/Gliederung</b>	<ul style="list-style-type: none"> <li>- Einführung in SAP Software</li> <li>- Navigation</li> <li>- Einführung in GBI</li> <li>- Vertrieb</li> <li>- Materialwirtschaft</li> <li>- Produktionsplanung und -steuerung</li> <li>- Finanzwesen</li> <li>- Controlling</li> <li>- Human Capital Management</li> <li>- Warehouse Management</li> <li>- Projektsystem</li> <li>- Integrierte Fallstudien</li> </ul>
<b>Qualifikationsziele</b>	Die Studierenden können in SAP ERP Software navigieren, Transaktionen aufrufen und buchen. Sie können betriebliche Daten durch Reports in SAP ERP Software analysieren. Sie haben das Integrationsmodell verstanden und können integrierte Fallstudien in SAP ERP Software bearbeiten.
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Kenntnisse Grundlagen der Betriebswirtschaft und Datenbanktechniken
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- CDI (Hrsg.): „SAP R/3® Einführung“, Pearson, 2001.</li> <li>- A. Maassen et al.: „Grundkurs SAP R/3®: Lern- und Arbeitsbuch“, Vieweg, 2003.</li> <li>- P. Wenzel: „Betriebswirtschaftliche Anwendungen mit SAP R/3“, Vieweg+Teubner, 1999.</li> <li>- T. Teufel et al.: „SAP-Prozesse, Finanzwesen und Controlling“, Addison-Wesley, 2000.</li> <li>- F. Klenger, E. Falk-Kalms: „Kostenstellenrechnung mit SAP R/3“, Vieweg, 2002.</li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	Keine Angabe
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Computeranimation Computer Animation
<b>Modulnummer</b>	C182 [INB8140, MIB8140] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Wintersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. Mario Hlawitschka <a href="mailto:mario.hlawitschka@htwk-leipzig.de">mario.hlawitschka@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. Mario Hlawitschka <a href="mailto:mario.hlawitschka@htwk-leipzig.de">mario.hlawitschka@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	5 ECTS-Punkte
<b>Workload</b>	150 Stunden
<b>Lehrveranstaltungen</b>	4 SWS (2 SWS Vorlesung   2 SWS Seminar)
<b>Selbststudienzeit</b>	94 Stunden 60 Stunden Bearbeitung Prüfungsleistung 34 Stunden Vorbereitung Lehrveranstaltung
<b>Prüfungsvorleistung(en)</b>	Keine
<b>Prüfungsleistung(en)</b>	Prüfung Projektarbeit Modulprüfung   Prüfungsdauer: 6 Wochen   Wichtigung: 100%
<b>Lehr- und Lernformen</b>	-
<b>Medienform</b>	keine Angabe
<b>Lehrinhalte/Gliederung</b>	<ul style="list-style-type: none"> <li>- Grundlagen der Computeranimation</li> <li>- Herstellung einer Computeranimation</li> <li>- Animationstechniken</li> <li>- Rendering</li> <li>- Erstellung von Spezialeffekten</li> </ul>

<b>Qualifikationsziele</b>	<p>Nach erfolgreichem Abschluss beherrschen die Studierenden Grundtechniken der 3D-Modellierung von Szenen mit Körpern als polygonale Netze, Prinzipien verschiedener Beleuchtungsverfahren und den Einsatz von Kameras. Sie beherrschen Verfahren der Computeranimation wie KeyframeAnimation, Methoden der inversen Kinematik, Motion Capture und Morphing. Durch Einsatz von Materialien und Mapping-Techniken sind sie in der Lage, die erstellten Szenen mit verschiedenen Renderverfahren fotorealistisch präsentieren.</p> <p>Die Studierenden setzen diese Kenntnisse in einem kommerziellen Computeranimationssystem bis zur Fertigstellung einer Computeranimation exemplarisch um. Sie sind in der Lage den Einsatz der Software für verschiedene Anwendungen einschätzen.</p>
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Grundlagen der Darstellenden Geometrie, Vorlesung Computergrafik (empfohlen), Programmierkenntnisse
<b>Literaturhinweise</b>	<p>Ein Skript oder Folien der Vorlesungen werden im OPAL zur Verfügung gestellt</p> <p>Ergänzende aktuelle Literatur zur Vorlesung findet sich im OPAL</p> <p>Die Vorlesung bezieht sich in Auszügen auf:</p> <p>A. H. Watt, M. Watt, "Advanced animation and rendering techniques: Theory and practice(Reprint.)", New York, NY, ACM Press, 1998.</p>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	Keine Angabe
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Diskrete Mathematik Discrete Mathematics
<b>Modulnummer</b>	N915 [INB8160, MIB8160] Version: 1
<b>Fakultät</b>	MNZ(Ma): Mathematik - Mathematisch-Naturwissenschaftliches Zentrum
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Wintersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. habil. Martin Grützmüller <a href="mailto:martin.gruettmueller@htwk-leipzig.de">martin.gruettmueller@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. habil. Martin Grützmüller <a href="mailto:martin.gruettmueller@htwk-leipzig.de">martin.gruettmueller@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	5 ECTS-Punkte
<b>Workload</b>	150 Stunden
<b>Lehrveranstaltungen</b>	4 SWS (2 SWS Vorlesung   2 SWS Seminar)
<b>Selbststudienzeit</b>	94 Stunden 32 Stunden Vorbereitung Lehrveranstaltung 42 Stunden Bearbeitung Prüfungsvorleistung 20 Stunden Vorbereitung Prüfung
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Beleg
<b>Prüfungsleistung(en)</b>	Prüfung Klausurarbeit Modulprüfung   Prüfungsdauer: 120 Minuten   Wichtigkeit: 100%   nicht kompensierbar
<b>Lehr- und Lernformen</b>	-
<b>Medienform</b>	keine Angabe

<b>Lehrinhalte/Gliederung</b>	<ul style="list-style-type: none"> <li>- Mengen und Relationen</li> <li>- Algebraische Strukturen (Modulare Arithmetik) und Ordnungsstrukturen</li> <li>- Graphentheorie <ul style="list-style-type: none"> <li>- Grundbegriffe</li> <li>- Paarungen, Packungen und Überdeckungen</li> <li>- Zusammenhang</li> <li>- Graphen in der Ebene</li> <li>- Färbungen</li> <li>- Flüsse</li> </ul> </li> </ul>
<b>Qualifikationsziele</b>	Nach erfolgreichem Abschluss des Moduls haben die Studierenden grundlegende Kenntnisse auf dem Gebiet diskreter mathematischer Strukturen erworben. Dazu gehört insbesondere das Erkennen und Klassifizieren von Algebraischen- und Ordnungsstrukturen. Die Studierenden besitzen vertiefte Kenntnisse der Graphentheorie, kennen Standardprobleme und können diese in geeigneten Anwendungsproblemen wiedererkennen. Die Studierenden können logische Argumentationen nachvollziehen und selber korrekt führen. Sie sind in der Lage Algorithmen zur Lösung von Aufgaben einzusetzen und selbständig zu entwickeln.
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Modellierungskompetenzen, Kompetenzen aus den Modulen Mathematik für Informatiker I und II
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- M. Aigner: "Diskrete Mathematik", Vieweg [ebook].</li> <li>- R. Diestel: "Graphentheorie", Springer Verlag, 2010.</li> <li>- V. Turan: "Algorithmische Graphentheorie", Oldenbourg Wissenschaftsverlag [ebook].</li> <li>- D. Jungnickel: "Graphen, Netzwerke und Algorithmen", BI-Wissenschaftsverlag, 1990.</li> <li>- D. Jungnickel: "Graphs, Networks und Algorithms", Springer, 2013.</li> <li>- Weitere aktuelle Literaturhinweise werden in den Lehrveranstaltungen gegeben.</li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	Keine Angabe
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Mobile Computing Mobile Computing
<b>Modulnummer</b>	C652 [INB8490, MIB8490, MIB-BI8490] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Sommersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. Uwe Petermann <a href="mailto:uwe.petermann@htwk-leipzig.de">uwe.petermann@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. Uwe Petermann <a href="mailto:uwe.petermann@htwk-leipzig.de">uwe.petermann@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	5 ECTS-Punkte
<b>Workload</b>	150 Stunden
<b>Lehrveranstaltungen</b>	4 SWS (2 SWS Vorlesung   2 SWS Seminar)
<b>Selbststudienzeit</b>	94 Stunden
<b>Prüfungsvorleistung(en)</b>	Keine
<b>Prüfungsleistung(en)</b>	Prüfung Computerarbeit Modulprüfung   Prüfungsdauer: 120 Minuten   Wichtigung: 100%
<b>Lehr- und Lernformen</b>	-
<b>Medienform</b>	keine Angabe
<b>Lehrinhalte/Gliederung</b>	<ul style="list-style-type: none"> <li>- Kommunikationsprotokolle für mobile Anwendungen.</li> <li>- Programmier-Plattformen für mobile Anwendungen (insbesondere Java Micro Edition, Android, IOS, weitere).</li> <li>- Techniken und Werkzeuge der Cross-Plattform-Entwicklung.</li> <li>- Sicherheitsaspekte bei Endgeräten, Kommunikation und Anwendungen</li> <li>- Praktische Übungen zur Konzeption und Realisierung von Anwendungen des Mobile Computing.</li> </ul>



<b>Qualifikationsziele</b>	Die Studierenden sind zur Konzeption und zur Entwicklung von Anwendungslösungen mit mobilen Kommunikationsgeräten der wichtigsten Plattformen befähigt. Sie beherrschen die aktuellen Standards und Kommunikationsprotokolle sowie die Programmierplattformen für mobile Endgeräte.
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Anwendungsbereite Kenntnisse zu Hard- und Software von Rechnern und Netzen; Beherrschung der Entwicklung von Lösungen für Praxisprobleme unter Verwendung höherer Programmiersprachen;  Befähigung zur Auswahl und zum Einsatz der für die Lösung von Praxisproblemen geeigneten Algorithmen und Datenstrukturen, sowie Werkzeuge.
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- M. Ross: "PhoneGap - Mobile Cross-Plattform-Entwicklung", dpunkt-Verlag, 2013.</li> <li>- J. Stark: "Building Android Apps with HTML, CSS, and JavaScript", O'Reilly, 2012.</li> <li>- U. Post: "Android-Apps entwickeln", Galileo Computing, 2012.</li> <li>- J. Roth: "Mobile Computing", dpunkt-Verlag, 2005.</li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	Keine Angabe
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Grundlagen der Künstlichen Intelligenz Foundations of Artificial Intelligence
<b>Modulnummer</b>	C622 [INB8012] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Sommersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. Sibylle Schwarz <a href="mailto:sibylle.schwarz@htwk-leipzig.de">sibylle.schwarz@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. Sibylle Schwarz <a href="mailto:sibylle.schwarz@htwk-leipzig.de">sibylle.schwarz@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	5 ECTS-Punkte
<b>Workload</b>	150 Stunden
<b>Lehrveranstaltungen</b>	4 SWS (2 SWS Vorlesung   2 SWS Seminar)
<b>Selbststudienzeit</b>	94 Stunden
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Beleg
<b>Prüfungsleistung(en)</b>	Prüfung Klausurarbeit Prüfungsdauer: 90 Minuten   Wichtigung: 100%
<b>Lehr- und Lernformen</b>	keine Angabe
<b>Medienform</b>	keine Angabe
<b>Lehrinhalte/Gliederung</b>	<ul style="list-style-type: none"> <li>- Intelligente Agenten: Aktionen und Verhalten, Struktur und Umgebungen</li> <li>- Repräsentation von Wissen: Logik, Regeln</li> <li>- Deduktion und Problemlösen</li> <li>- Ausblick logische Programmierung, Resolution</li> <li>- Suchverfahren</li> <li>- Wissensbasiertes Planen</li> <li>- Ausgewählte Beispiele: Robotik, Spiele und Diagnosesysteme</li> <li>- Ausblick nichtklassische Logiken: nichtmonotones Schließen, Temporallogik, Fuzzy-Logik</li> </ul>

<b>Qualifikationsziele</b>	Die Studierenden kennen Grundlagen und praktische Anwendungen der Wissensverarbeitung und der Künstlichen Intelligenz. Sie können basierend auf den Kenntnissen zu ausgewählten Formen der Darstellung von Wissen und zu Problemlösungsverfahren einfache Probleme aus dem Bereich der KI analysieren und lösen.
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Inhalte des Moduls „Modellierung“
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- I. Boersch, J. Heinsohn, R. Socher-Ambrosius: „Wissensverarbeitung. Eine Einführung in die Künstliche Intelligenz für Informatiker und Ingenieure“, Spektrum Akademischer Verlag, 2007.</li> <li>- S. Russell, P. Norvig: „Künstliche Intelligenz“, Pearson, 2012.</li> <li>- C. Beierle, G. Kern-Isberner: „Methoden wissensbasierter Systeme“, Vieweg, 2006.</li> <li>- W. Ertel: „Grundkurs Künstliche Intelligenz“, Vieweg, 2008.</li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	Keine Angabe
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Computergrafik Computer Graphics
<b>Modulnummer</b>	C121 [MIB4030, INB8013, MIB-BI8470] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Sommersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. Mario Hlawitschka <a href="mailto:mario.hlawitschka@htwk-leipzig.de">mario.hlawitschka@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. Mario Hlawitschka <a href="mailto:mario.hlawitschka@htwk-leipzig.de">mario.hlawitschka@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	5 ECTS-Punkte
<b>Workload</b>	150 Stunden
<b>Lehrveranstaltungen</b>	4 SWS (2 SWS Vorlesung   2 SWS Seminar)
<b>Selbststudienzeit</b>	94 Stunden 20 Stunden Vorbereitung Prüfung 22 Stunden Vorbereitung Lehrveranstaltung 52 Stunden Bearbeitung Prüfungsvorleistung
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Prüfung Computerarbeit
<b>Prüfungsleistung(en)</b>	Prüfung Klausurarbeit Modulprüfung   Prüfungsdauer: 120 Minuten   Wichtigkeit: 100%   nicht kompensierbar
<b>Lehr- und Lernformen</b>	-
<b>Medienform</b>	keine Angabe
<b>Lehrinhalte/Gliederung</b>	<ul style="list-style-type: none"> <li>- Gerätetechnik</li> <li>- Algorithmen der Computergrafik</li> <li>- Geometrische Transformationen</li> <li>- Visualisierung</li> <li>- Datenmodelle für geometrische Objekte</li> </ul>

<b>Qualifikationsziele</b>	<p>Nach erfolgreichem Abschluss der Lehrveranstaltung sind die Studierenden in der Lage Methoden der generativen Computergrafik wie Modellierung, Transformation und Visualisierung von geometrischen Objekten in Projekten einzusetzen.</p> <p>Sie können die Stärken und Schwächen der geometrischen Modelle sowie ihre Einsatzmöglichkeiten einschätzen und beherrschen die entsprechenden mathematischen Grundlagen.</p>
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Programmieren in einer Objektorientierten Programmiersprache, Analytische Geometrie, Lineare Algebra
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- Ein Skript oder Folien der Vorlesungen werden in OPAL zur Verfügung gestellt.</li> <li>- Ergänzende aktuelle Literatur zur Vorlesung findet sich in OPAL.</li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	Prüfungsvorleistung am Computer (PVC): Bearbeitung einer Praktikumsaufgabe und Präsentation der Ergebnisse am Computer.
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Audio-Video-Kommunikation Audio-Video Communication
<b>Modulnummer</b>	C064 [INB8014] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Wintersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr.-Ing. Jean-Alexander Müller <a href="mailto:jean-alexander.mueller@htwk-leipzig.de">jean-alexander.mueller@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr.-Ing. Jean-Alexander Müller <a href="mailto:jean-alexander.mueller@htwk-leipzig.de">jean-alexander.mueller@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	5 ECTS-Punkte
<b>Workload</b>	150 Stunden
<b>Lehrveranstaltungen</b>	4 SWS (2 SWS Übung   2 SWS Praktikum)
<b>Selbststudienzeit</b>	94 Stunden 44 Stunden Vorbereitung Lehrveranstaltung 50 Stunden Bearbeitung Prüfungsleistung
<b>Prüfungsvorleistung(en)</b>	Keine
<b>Prüfungsleistung(en)</b>	Prüfung mündliches Fachgespräch Prüfungsdauer: 25 Minuten   Wichtigung: 100%
<b>Lehr- und Lernformen</b>	-
<b>Medienform</b>	-
<b>Lehrinhalte/Gliederung</b>	<ul style="list-style-type: none"> <li>- Technologische Voraussetzungen</li> <li>- Bedingungen für die multimediale Kommunikation</li> <li>- Kommunikationsmodelle und -dienste</li> <li>- Multimedia – Digitalisierung, Codecs, Präsentation, Systemaufbau</li> <li>- Netzwerk-Technologien für multimediale Kommunikation</li> <li>- Multimediale Kommunikation</li> <li>- Multimediale Anwendungen</li> </ul>

<b>Qualifikationsziele</b>	Ziele: detailliertes Fachwissen auf dem Gebiet der multimedialen Kommunikation, zu ihren Einsatzcharakteristika, zu deren Nutzung und zu den Bedingungen / Voraussetzungen eines effektiven Einsatzes detailliertes praxisrelevantes Fachwissen zu einer ausgewählten Spezialrichtung
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Betriebssysteme, Rechnernetze und Programmierung (C)
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- P. L. Dordal: "An Introduction to Computer Networks", ebook, 2018.</li> <li>- R. Steinmetz, K. Nahrstedt: „Multimedia Systems“, Springer 2004.</li> <li>- R. Steinmetz, K. Nahrstedt: „Multimedia Applications“, Springer 2004.</li> <li>- C. Meinel, H. Sack, „Digitale Kommunikation: Vernetzen, Multimedia, Sicherheit“, Springer, 2009.</li> <li>- R. Steinmetz: „Multimedia-Technologie: Grundlagen, Komponenten und Systeme“, Springer, 2000.</li> <li>- W. Effelsberg, R. Steinmetz: „Video Compression Techniques. From JPEG to Wavelets“, dpunkt, 2001.</li> <li>- T. Milde: „Videokompressionsverfahren im Vergleich. JPEG, MPEG, H.261, XCCC, Wavelets, Fraktale“, dpunkt, 1999.</li> <li>- K. Froitzheim: „Multimedia-Kommunikation Dienste, Protokolle und Technik für Telekommunikation und Computernetze“, dpunkt, 1997.</li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	Keine Angabe
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Datenbanken (Aufbaukurs) Database Systems (advanced level)
<b>Modulnummer</b>	C720 [MIB-BI5050, INB8150, MIB8150] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Wintersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr.-Ing. Thomas Kudraß <a href="mailto:thomas.kudrass@htwk-leipzig.de">thomas.kudrass@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr.-Ing. Thomas Kudraß <a href="mailto:thomas.kudrass@htwk-leipzig.de">thomas.kudrass@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	5 ECTS-Punkte
<b>Workload</b>	150 Stunden
<b>Lehrveranstaltungen</b>	4 SWS (2 SWS Vorlesung   2 SWS Seminar)
<b>Selbststudienzeit</b>	94 Stunden
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Testat
<b>Prüfungsleistung(en)</b>	Prüfung mündliches Fachgespräch Prüfungsdauer: 30 Minuten   Wichtigung: 100%
<b>Lehr- und Lernformen</b>	-
<b>Medienform</b>	keine Angabe
<b>Lehrinhalte/Gliederung</b>	<ul style="list-style-type: none"> <li>- Datenbank-Anwendungsprogrammierung mit PL/SQL (Oracle)</li> <li>- Objektrelationale und objektorientierte Datenbanken</li> <li>- XML und Datenbanken (Speicherung von XML, Anfragesprachen: XML/SQL, XQuery)</li> <li>- Java und Datenbanken (JDBC, Hibernate)</li> <li>- NoSQL-Datenbanken</li> <li>- Datenbanken im Web (Anwendungen, Systemarchitekturen, DB-Zugriffsschnittstellen)</li> </ul>



<b>Qualifikationsziele</b>	Nach erfolgreichem Abschluss des Moduls hat der Student umfangreiche Erfahrungen bei der Entwicklung von Datenbankprojekten. Er kann die Konzepte einer Datenbankprogrammiersprache bei der Lösung von praktischen Programmieraufgaben anwenden. Der Student kennt eine Reihe von Datenbankmodellen, die das Relationenmodell erweitern bzw. alternativ dazu gesehen werden können und kann deren Merkmale für bestimmte Anwendungen bewerten. Der Student benutzt eine Vielzahl von Datenbankzugriffsschnittstellen mit unterschiedlichem Abstraktionsniveau bei Programmierübungen. Er ist in der Lage, die Vor- und Nachteile von unterschiedlichen Zugriffsschnittstellen bzw. Datenbankmodellen einzuschätzen. Mit diesem gewonnenen Wissen wird der Student befähigt, bei der Entwicklung eines datenbankbasierten Informationssystems eine geeignete Systemarchitektur zu entwerfen und die Anforderungen der jeweiligen Anwendung zu berücksichtigen. Schwerpunktmäßig wird dieses Wissen auf die Entwicklung von Datenbanken im Web angewendet.
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Der Student beherrscht einen Datenbankentwurf und kann einfache Anfragen mittels SQL formulieren.
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- M. Skulschus, M. Wiederstein: "Oracle, PL/SQL und XML", Comelio Medien, in der aktuellen Auflage.</li> <li>- H. Wehr, B. Müller: "Java Persistence API 2: Hibernate, EclipseLink, OpenJPA und Erweiterungen", Carl Hanser Verlag, 2012.</li> <li>- S. Edlich et al.: "NoSQL: Einstieg in die Welt nichtrelationaler Web 2.0 Datenbanken", Carl Hanser Verlag, in der aktuellen Auflage.</li> <li>- Weitere aktuelle Literaturhinweise unter <a href="http://www.kudrass.de">www.kudrass.de</a></li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	Keine Angabe
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Multimediale Webprogrammierung Multimedia Web Programming
<b>Modulnummer</b>	C741 [MIB4020, MIB-BI4020, INB8021] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Sommersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. habil. Michael Frank <a href="mailto:michael.frank@htwk-leipzig.de">michael.frank@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. habil. Michael Frank <a href="mailto:michael.frank@htwk-leipzig.de">michael.frank@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	5 ECTS-Punkte
<b>Workload</b>	150 Stunden
<b>Lehrveranstaltungen</b>	4 SWS (2 SWS Vorlesung   2 SWS Seminar)
<b>Selbststudienzeit</b>	94 Stunden 30 Stunden Sonstiges 64 Stunden Selbststudium
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Beleg
<b>Prüfungsleistung(en)</b>	Prüfung Klausurarbeit Modulprüfung   Prüfungsdauer: 120 Minuten   Wichtigkeit: 100%   nicht kompensierbar
<b>Lehr- und Lernformen</b>	-
<b>Medienform</b>	keine Angabe
<b>Lehrinhalte/Gliederung</b>	<ul style="list-style-type: none"> <li>- HTML5 und seine Strukturelemente, Dokumentstrukturierung</li> <li>- Grundlagen des CSS-Stylings, Boxendesign, Schatten, Farbverläufe, Transparenzen, Transformationen, SVG-Nutzung</li> <li>- Nutzung von JavaScript und von JavaScript-Bibliotheken wie jQuery</li> <li>- Spezialaspekte wie Canvas, Drag&amp;Drop, Geolocation, Storage, File, Audio und Video, u.a..</li> <li>- Weitere Aspekte ja nach Entwicklungen rund um HTML5.</li> <li>- Praktische Übungen aller Aspekte.</li> </ul>

<b>Qualifikationsziele</b>	<p>Die Studierenden beherrschen moderne Cross-Plattform-Webprogrammierung mit HTML5, CSS3, Web APIs und JavaScript-Bibliotheken unter Berücksichtigung von Aspekten unterschiedlicher Webbrowser.</p> <p>Sie sind mit Prinzipien der Barrierefreiheit in der Webprogrammierung vertraut und befähigt, sich mit der weiteren dynamischen Entwicklung der Webprogrammierung selbständig auseinanderzusetzen.</p>
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Kompetenzen in statischer Webprogrammierung mit HTML, CSS und JavaScript einschließlich DOM
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- J. D. Gauchat: "HTML5, CSS3 und JavaScript", Wiley-VCH, 2013.</li> <li>- M. Vollendorf, F. Bongers: "jQuery. Das Praxisbuch.", Galileo Press, 2011.</li> <li>- F. Franke, J. Ippen: "Apps mit HTML5 und CSS3. Für iPhone, iPad und Android.", Galileo Press, 2012.</li> <li>- Div. Schriftquellen und Internetquellen je nach Thematik und Zeitraum.</li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	<p>Sonstige Selbststudienzeit: Projekt 30 h</p> <p>Prüfungsvorleistung Belege (PVB): Übungsfragen und praktische Übungsaufgaben</p>
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Assemblerprogrammierung Assembler Programming
<b>Modulnummer</b>	C960 [INB8022] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Sommersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. Heinrich Krämer <a href="mailto:heinrich.kraemer@htwk-leipzig.de">heinrich.kraemer@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. Heinrich Krämer <a href="mailto:heinrich.kraemer@htwk-leipzig.de">heinrich.kraemer@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	5 ECTS-Punkte
<b>Workload</b>	150 Stunden
<b>Lehrveranstaltungen</b>	4 SWS (2 SWS Vorlesung   2 SWS Seminar)
<b>Selbststudienzeit</b>	94 Stunden 60 Stunden Bearbeitung Prüfungsleistung 34 Stunden Vorbereitung Lehrveranstaltung
<b>Prüfungsvorleistung(en)</b>	Keine
<b>Prüfungsleistung(en)</b>	Prüfung Projektarbeit Prüfungsdauer: 6 Wochen   Wichtigkeit: 100%   nicht kompensierbar
<b>Lehr- und Lernformen</b>	keine Angabe
<b>Medienform</b>	keine Angabe

<b>Lehrinhalte/Gliederung</b>	<ul style="list-style-type: none"> <li>- i486-Programmiermodell im Real address mode</li> <li>- Adressierungsarten</li> <li>- Einsatz verschiedener Assemblerbefehle</li> <li>- Unterprogramme, Parameterübergabetechniken</li> <li>- Interrupt-Verarbeitung</li> <li>- Gleitpunkt-Einheit</li> <li>- MXX-, SSE(II)-Einheit</li> <li>- Protected mode, Schutzkonzepte, Hardwareunterstützung für Systemprogrammierung</li> </ul>
<b>Qualifikationsziele</b>	Die Studenten sollen die Möglichkeiten kennen und beherrschen, Programme durch Ausnutzung der Prozessorarchitektur zu optimieren. Die Studenten können mit typischen Problemen bei der hardwarenahen Programmierung umgehen.
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Es werden Grundkenntnisse in der Programmierung und Rechnerarchitektur vorausgesetzt
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- J. Erdweg: „Assemblerprogrammierung mit dem PC“, Vieweg, 1992.</li> <li>- T. E. Podschun: „Das Assemblerbuch“, Addison-Wesley, 2002.</li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	Keine Angabe
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Sprachkonzepte der parallelen Programmierung Language Concepts for Parallel Programming
<b>Modulnummer</b>	C529 [INB8023] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Wintersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. Johannes Waldmann <a href="mailto:johannes.waldmann@htwk-leipzig.de">johannes.waldmann@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. Johannes Waldmann <a href="mailto:johannes.waldmann@htwk-leipzig.de">johannes.waldmann@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	5 ECTS-Punkte
<b>Workload</b>	150 Stunden
<b>Lehrveranstaltungen</b>	4 SWS (2 SWS Vorlesung   2 SWS Seminar)
<b>Selbststudienzeit</b>	94 Stunden
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Beleg
<b>Prüfungsleistung(en)</b>	Prüfung Klausurarbeit Prüfungsdauer: 120 Minuten   Wichtung: 100%   nicht kompensierbar
<b>Lehr- und Lernformen</b>	keine Angabe
<b>Medienform</b>	keine Angabe
<b>Lehrinhalte/Gliederung</b>	<ul style="list-style-type: none"> <li>- Abstraktionen zur Thread-Synchronisation und -Kommunikation</li> <li>- thread-sichere Collections-Datentypen</li> <li>- spekulative Ausführung (Software Transactional Memory)</li> <li>- Rekursionsschemata für parallele funktionale Programme, map/reduce</li> </ul>
<b>Qualifikationsziele</b>	Student kennt Ausdrucksmittel für parallele und nebenläufige Programme in verschiedenen Programmierparadigmen und –sprachen und kann diese anwenden. Student kann Aussagen über Korrektheit und Ressourcenverbrauch formulieren und begründen.

<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	keine
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- B. Goetz et al.: „Java Concurrency in Practice“, Addison Wesley, 2006.</li> <li>- Herlihy, M., Shavit, N.: „The Art of Multiprocessor Programming“, Morgan Kaufmann, 2008.</li> <li>- Hoare, C.A.R.: „Communicating Sequential Processes“, Prentice Hall, 2004</li> <li>- Peyton Jones, S.: „Beautiful Concurrency“, in: Wilson, G. (Hrsg.): „Beautiful Code“, O'Reilly, 2007.</li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	Keine Angabe
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Systemprogrammierung Systems Programming
<b>Modulnummer</b>	C530 [INB8024] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Wintersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. Jens Wagner <a href="mailto:jens.wagner@htwk-leipzig.de">jens.wagner@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. Klaus Bastian <a href="mailto:klaus.bastian@htwk-leipzig.de">klaus.bastian@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	5 ECTS-Punkte
<b>Workload</b>	150 Stunden
<b>Lehrveranstaltungen</b>	4 SWS (2 SWS Vorlesung   2 SWS Praktikum)
<b>Selbststudienzeit</b>	94 Stunden 60 Stunden Bearbeitung Prüfungsvorleistung 34 Stunden Vorbereitung Lehrveranstaltung
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Projekt
<b>Prüfungsleistung(en)</b>	Prüfung mündliches Fachgespräch Prüfungsdauer: 30 Minuten   Wichtigung: 100%   nicht kompensierbar
<b>Lehr- und Lernformen</b>	keine Angabe
<b>Medienform</b>	keine Angabe



<b>Lehrinhalte/Gliederung</b>	<ul style="list-style-type: none"> <li>- UNIX und sein E/A-Subsystem</li> <li>Betriebssystemkern, Prozesssteuerung und -zeitplanung, Systemaufrufe</li> <li>- Das UNIX E/A-System</li> <li>Dateisystem, Systemdatenstrukturen, Blockpuffersystem, Device-Treiber</li> <li>- E/A-Hardware</li> <li>Bussysteme, Adressierungsarten, Pufferspeicher, Interrupts, programmierte E/A, DMA</li> <li>- Systemgenerierung</li> <li>Treibermodule, Modulschnittstellen, Einfügen und Registrieren von Treibern und ihren Ressourcen</li> </ul>
<b>Qualifikationsziele</b>	Die Studenten haben ein präzises Verständnis von den Funktionen des Betriebssystemkerns und den Wechselwirkungen bezüglich Zeitplanung und Ablaufsteuerung. Sie haben modellhafte Vorstellungen von der Modulschnittstelle eines Betriebssystems und sind in der Lage, für verschiedene Gerätetypeneinfache Kernelmodule zu entwerfen, zu implementieren, zu testen und zu dokumentieren. Insbesondere können Sie die Hard- und Softwarekomplexität von Geräten und Treibern abschätzen.
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Die Studenten beherrschen verschiedene Kontrollstrukturen von den Zustandsfolgen endlicher Automaten bis zum Timesharing in Interruptsystemen und sie können damit Aufgabenstellungen in verteilten und zeitlich parallelen Anwendungen lösen.
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- J. I. Egan, T. J. Teixera: „UNIX Device-Treiber“, Addison-Weseley, 1990.</li> <li>- E.-K. Kunst, J. Quade: „Kern-Technik“, Linux-Magazin, Artikelfolge 2013ff.</li> <li>- J. Corbet et al.: „Linux Device Drivers“, O'Reilly, 2005.</li> <li>- A. S. Tanenbaum: „Moderne Betriebssysteme“, Pearson, 2009</li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	Keine Angabe
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Grundlagen Internet-basierter Informationssysteme Introduction to Internet Based Information Systems
<b>Modulnummer</b>	C247 [INB8031, MIB8031, MIB-BI8410] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Wintersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. Thomas Riechert <a href="mailto:thomas.riechert@htwk-leipzig.de">thomas.riechert@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. Thomas Riechert <a href="mailto:thomas.riechert@htwk-leipzig.de">thomas.riechert@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	5 ECTS-Punkte
<b>Workload</b>	150 Stunden
<b>Lehrveranstaltungen</b>	4 SWS (2 SWS Vorlesung   2 SWS Übung)
<b>Selbststudienzeit</b>	94 Stunden
<b>Prüfungsvorleistung(en)</b>	Keine
<b>Prüfungsleistung(en)</b>	Prüfung mündliches Fachgespräch Prüfungsdauer: 30 Minuten   Wichtigung: 100%   nicht kompensierbar
<b>Lehr- und Lernformen</b>	keine Angabe
<b>Medienform</b>	keine Angabe

<b>Lehrinhalte/Gliederung</b>	<ul style="list-style-type: none"> <li>- Einführung – Geschichte und Struktur des Internets</li> <li>- Einführung – Verteilte Informationssysteme</li> <li>- Internet-Stack, Infrastruktur (Router, Netztopographien)</li> <li>- Applikationsschicht (Ausgewählte Anwendungen)</li> <li>- HTTP-Protokoll / http-Webarchitekturen</li> <li>- Web-Architekturen (allgemein)</li> <li>- Service Orientierte Architekturen (SOA), Webservices</li> <li>- JSON-REST-Webschnittstelle</li> <li>- Semantic Web</li> <li>- Verteilte Informationsverarbeitung</li> <li>- Im Rahmen der Übung werden die Inhalte der Vorlesung in praktischen Experimenten nachvollzogen.</li> </ul> <p>Dabei werden u.a. ein Unix-Server installiert, verschiedene Webapplikationen installiert und getestet, sowie Schnittstellen definiert und entwickelt</p>
<b>Qualifikationsziele</b>	Nach erfolgreichem Besuch der Veranstaltung sind die Studierenden in der Lage die Protokolle und Systemkomponenten für die Kommunikation paralleler Prozesse über Internetverbindungen zu beurteilen und auszuwählen. Sie können damit auf der Basis von TCP und UDP komplexe verteilte Anwendungen und Schnittstellen für Internet-basierte Informationssysteme entwickeln.
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Die Studierenden beherrschen den Aufbau und die Arbeitsweise von Rechnernetzen und die darin eingesetzten Protokollhierarchien.
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- Ch. Meinel, H. Sack: „Internetworking: Technische Grundlagen und Anwendungen“, Springer, 2012.</li> <li>- A. S. Tanenbaum, D. Wetherall: „Computernetzwerke“, Pearson, 2012.</li> <li>- Weiterführende Literaturhinweise werden zu Beginn der Lehrveranstaltung bekanntgegeben.</li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	Keine Angabe
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Prozessautomatisierung Automation of Technical Processes
<b>Modulnummer</b>	C383 [INB8032] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Sommersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr.-Ing. Axel Schneider <a href="mailto:axel.schneider@htwk-leipzig.de">axel.schneider@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr.-Ing. Axel Schneider <a href="mailto:axel.schneider@htwk-leipzig.de">axel.schneider@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	5 ECTS-Punkte
<b>Workload</b>	150 Stunden
<b>Lehrveranstaltungen</b>	4 SWS (2 SWS Vorlesung   2 SWS Praktikum)
<b>Selbststudienzeit</b>	94 Stunden
<b>Prüfungsvorleistung(en)</b>	Prüfungsvorleistung Beleg
<b>Prüfungsleistung(en)</b>	Prüfung mündliches Fachgespräch Prüfungsdauer: 30 Minuten   Wichtung: 100%
<b>Lehr- und Lernformen</b>	keine Angabe
<b>Medienform</b>	keine Angabe
<b>Lehrinhalte/Gliederung</b>	<ul style="list-style-type: none"> <li>- Automatisierung technischer Prozesse</li> <li>- Stetige und binäre Steuerungen</li> <li>- Speicherprogrammierbare Steuerungen</li> <li>- Regelungen und Fuzzy Control</li> <li>- Neuronale Konzepte und Neuro-Fuzzy-Control</li> </ul>

<b>Qualifikationsziele</b>	Die Studierenden können nach dem erfolgreichen Abschluss des Moduls, bestimmte technische Prozesse durch den Einsatz von informationsverarbeitenden Systemen automatisieren. Dazu werden insbesondere für verschiedene Aufgabenklassen Steuerungen und Regelungen entworfen und diese in entsprechende Programme umgesetzt und getestet. Dabei kommen insbesondere SPSe zum Einsatz, auf deren Grundlage verschiedene Programmierungsmöglichkeiten genutzt werden.
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Kenntnisse und Fertigkeiten zum Entwurf von Schaltnetzen und Schaltwerken
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- R. Langmann: „Taschenbuch der Automatisierung“, Fachbuchverlag Leipzig, aktuelle Auflage.</li> <li>- R. Lauber, P. Göhner: „Prozessautomatisierung“, Springer, aktuelle Auflage.</li> <li>- M. Seitz: „Speicherprogrammierbare Steuerungen“, Fachbuchverlag Leipzig, aktuelle Auflage.</li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	Keine Angabe
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Digitale Signalverarbeitung Digital Signal Processing
<b>Modulnummer</b>	C807 [INB8033] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Sommersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. Heinrich Krämer <a href="mailto:heinrich.kraemer@htwk-leipzig.de">heinrich.kraemer@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. Heinrich Krämer <a href="mailto:heinrich.kraemer@htwk-leipzig.de">heinrich.kraemer@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	5 ECTS-Punkte
<b>Workload</b>	150 Stunden
<b>Lehrveranstaltungen</b>	4 SWS (2 SWS Vorlesung   2 SWS Seminar)
<b>Selbststudienzeit</b>	94 Stunden
<b>Prüfungsvorleistung(en)</b>	Keine
<b>Prüfungsleistung(en)</b>	Prüfung mündliches Fachgespräch Prüfungsdauer: 30 Minuten   Wichtigung: 100%   nicht kompensierbar
<b>Lehr- und Lernformen</b>	keine Angabe
<b>Medienform</b>	keine Angabe
<b>Lehrinhalte/Gliederung</b>	<ul style="list-style-type: none"> <li>- Analoge ,digitale Signale, Filtertypen</li> <li>- LTI-Systeme: Impulsantwort, Kausalität, Stabilität</li> <li>- Transformationen</li> <li>• Fourier-Transformation, Abtasttheorem</li> <li>• Diskrete Fourier-Transformation</li> <li>• z-Transformation</li> <li>- Entwurf digitaler Filter</li> <li>• FIR-Filter: Fenstertechnik, Frequenzabtastung, Equiripple design</li> <li>• IIR-Filter: Typen analoger Filter, Bilineare Transformation, Realisierung (Biquad)</li> <li>- Fast Fourier Transformation (FFT)</li> </ul>

<b>Qualifikationsziele</b>	Die Studenten sind vertraut mit dem Entwurf und der Implementierung von grundlegenden Funktionen der digitalen Signalverarbeitung. Sie können grundlegende Algorithmen der DSV entwerfen und bewerten
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	keine
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- E. Ifeachor, J. Barrie: „Digital Signal Processing: A Practical Approach“, Addison Wesley, 2001.</li> <li>- A. V. Oppenheim et al.: „Zeitdiskrete Signalverarbeitung“, Addison-Wesley, 2004.</li> <li>- L. R. Rabiner: „Theory and Application of Digital Signal Processing“, Prentice Hall, 1975.</li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	Keine Angabe
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

<b>Modul</b>	Mikroprogrammierung und Mikroprozessoren Microprogramming and Microprocessors
<b>Modulnummer</b>	C004 [INB8034] Version: 1
<b>Fakultät</b>	FIM(I): Fakultät Informatik und Medien III (Informatik)
<b>Niveau</b>	Bachelor
<b>Dauer</b>	1 Semester
<b>Turnus</b>	Sommersemester
<b>Modulverantwortliche/-r</b>	Prof. Dr. rer. nat. Jens Wagner <a href="mailto:jens.wagner@htwk-leipzig.de">jens.wagner@htwk-leipzig.de</a>
<b>Dozent/-in(nen)</b>	Prof. Dr. rer. nat. Jens Wagner <a href="mailto:jens.wagner@htwk-leipzig.de">jens.wagner@htwk-leipzig.de</a>
<b>Sprache(n)</b>	Deutsch
<b>ECTS-Leistungspunkte</b>	5 ECTS-Punkte
<b>Workload</b>	150 Stunden
<b>Lehrveranstaltungen</b>	4 SWS (2 SWS Vorlesung   2 SWS Praktikum)
<b>Selbststudienzeit</b>	94 Stunden
<b>Prüfungsvorleistung(en)</b>	Keine
<b>Prüfungsleistung(en)</b>	Prüfung Projektarbeit Prüfungsdauer: 3 Monate   Wichtung: 100%   nicht kompensierbar
<b>Lehr- und Lernformen</b>	keine Angabe
<b>Medienform</b>	keine Angabe
<b>Lehrinhalte/Gliederung</b>	<ul style="list-style-type: none"> <li>- Hardwarebeschreibungssprachen für kombinatorische und sequenzielle Systeme</li> <li>- Automaten, Mikroprogrammierung und Mikroprogrammsteuerwerke</li> <li>- Mikroprogrammsteuerwerk und Hardwaresteuerwerk im Vergleich: Verschiedene Automatentypen, Minimierung des Aufwandes für den Mikroprogrammspeicher, Ein mikroprogrammierbarer Rechner</li> <li>- Mikroprozessoren und Mikrorechner: Zeitverhalten, Adressierungsarten, Befehlsausführung, Interruptsystem, Periphere Systembauelemente</li> </ul>



<b>Qualifikationsziele</b>	Die Studenten sind in der Lage, die verschiedenen Architekturprinzipien mikroelektronischer Systeme zu charakterisieren und typische Anwendungen mit den hierfür geeigneten Hard- und SoftwareWerkzeugen zu implementieren. Die Studenten beherrschen verschiedene Kontrollstrukturen von den Zustandsfolgen endlicher Automaten bis zum Timesharing in Interruptsystemen. Sie können damit Aufgabenstellungen in verteilten und zeitlich parallelen Anwendungen implementieren. Insbesondere sind die Voraussetzungen geschaffen, sich mit Kernel- und Treiberprogrammierung auseinanderzusetzen.
<b>Zulassungsvoraussetzung</b>	Keine
<b>Empfohlene Voraussetzungen</b>	Sicherer Umgang mit elektrischen Grundgrößen, ihren Maßeinheiten und den typischen Messmitteln. Handhabung grundlegender Methoden des Logikentwurfs kombinatorischer Funktionen und endlicher Automaten sowie deren Test in Simulationsumgebungen und in Hardwareanwendungen.
<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>- R. Hoffmann: „Rechnerentwurf: Rechenwerke, Mikroprogrammierung, RISC“, Oldenbourg, 1993.</li> <li>- H. Kieser, M. Meder: „Mikroprozessortechnik“, Verlag Technik, 1982.</li> <li>- D. Patterson, J. L. Hennessy: „Rechnerorganisation und Rechnerentwurf: Die Hardware/SoftwareSchnittstelle“, Oldenbourg, 2011.</li> </ul>
<b>Aktuelle Lehrressourcen</b>	keine
<b>Hinweise</b>	Keine Angabe
<b>Link zu Kurs/Lernressourcen im OPAL/Moodle/etc.</b>	

# HITWK

Hochschule für Technik,  
Wirtschaft und Kultur Leipzig

Fakultät  
Informatik und Medien

## **Praktikumsordnung**

für die  
Bachelorstudiengänge

## **Informatik Medieninformatik**

an der Hochschule für Technik, Wirtschaft und Kultur (HTWK) Leipzig

**(PraktO-B IN MI)**

## **§ 1 Geltungsbereich**

Diese Ordnung gilt für die Studierenden der Bachelorstudiengänge Informatik und Medieninformatik an der Fakultät Informatik und Medien der Hochschule für Technik, Wirtschaft und Kultur Leipzig.

## **§ 2 Inhalt**

Diese Ordnung ist ergänzender Teil der Studienordnungen der Bachelorstudiengänge Informatik und Medieninformatik (StudO-INB bzw. StudO-MIB). Die Praxisphase beinhaltet die Bearbeitung eines Praxisprojektes und die Anfertigung der Bachelorarbeit. Diese Ordnung betrifft allein das Praxisprojekt.

Für ein Praxisprojekt im Ausland, das seitens der Hochschule besonders gefördert wird, gilt diese Ordnung analog.

## **§ 3 Ziel des Praxisprojekts**

Das Praxisprojekt ist als integrierter Bestandteil des Studiums grundsätzlich dem Ausbildungsziel des Studiengangs INB bzw. MIB untergeordnet. Das Praxisprojekt hat insbesondere das Ziel, eine enge Verbindung zwischen Studium und Berufspraxis herzustellen und die Studierenden in die Berufswirklichkeit zu versetzen. Dabei sollen die Studierenden ihren eigenen theoretischen Kenntnisstand anhand der berufsspezifischen Praxisanforderungen überprüfen und ableiten, wo und in welcher Richtung sie ihr theoretisches Wissen vertiefen und erweitern müssen. Gleichzeitig können die Studenten ihre besonderen Neigungen, Fähigkeiten und Fertigkeiten mit den Anforderungen einzelner Tätigkeitsbereiche vergleichen und damit die Wahl ihres künftigen Einsatzes nach Studienabschluss mit größerer Sicherheit treffen. Ebenso soll das Praxisprojekt zur Vertiefung sozialer Kompetenzen beitragen.

## **§ 4 Einsatzgebiete**

(1) Das Praxisprojekt umfasst die Bearbeitung einer Schwerpunktaufgabe in einem IT-Projekt. Als Tätigkeiten kommen beispielsweise in Frage:

- Kommerzielle oder wissenschaftlich-technische Anwendungsprogrammierung
- Systemprogrammierung (Betriebssysteme, Compiler)
- Programmierung von (multimedialen) Informationssystemen, Datenbankanwendungen und Informationsvisualisierungen
- Programmierung von Anwendungen für mobile Geräte
- Entwicklung, Adaption und Einsatz von Content Management Systemen
- Programmierung von E-Learning-Systemen
- Mediengestaltung oder digitale Spieleentwicklung
- Entwicklung von CAD-Systemen
- Hardwareentwicklung
- Administration von Rechnernetzen

- Evaluation und Bewertung von Softwaresystemen
- Entwurf von Anwendungskonzepten und Einsatzvorbereitung von IT-Systemen

(2) Nicht als Praxisprojekt anerkannt werden beispielsweise:

- Tätigkeit auf Messen und Ausstellungen
- Verkaufs- und Vertriebstätigkeit
- Anwendungsberatung zum Einsatz von Standardsoftware
- Kurzzeitige Anwenderschulung
- Reine Literaturstudien

(3) Die Praxisphase kann in Unternehmen der gewerblichen Wirtschaft, des Dienstleistungsbereiches, in Institutionen der öffentlichen Hand und in Forschungseinrichtungen absolviert werden.

## **§ 5**

### **Umfang und Zeiträume**

(1) Das Praxisprojekt umfasst mindestens 12 Wochen praktische Tätigkeit im Berufsfeld (Vollzeit-tätigkeit). Dabei werden den Studenten in geeigneten Ausbildungsstellen – nachfolgend Praxisstellen genannt – praktische Erfahrungen und Kenntnisse zur Ergänzung der theoretischen Ausbildung vermittelt.

(2) Für die Praxisphase ist das 6. Fachsemester vorgesehen. Das Praxisprojekt muss bis spätestens 2 Wochen vor dem Ende der Vorlesungszeit im Sommersemester abgeschlossen sein. Über Ausnahmen entscheidet der Praktikumsbeauftragte des jeweiligen Studienganges.

## **§ 6**

### **Zulassung**

(1) Die Zulassung zur Praxisphase setzt in der Regel das Bestehen aller in der Studienordnung für die ersten drei Fachsemester vorgesehenen Prüfungen voraus. Über Ausnahmen entscheidet der Prüfungsausschuss auf Antrag des Studenten unter Einbeziehung des Praktikumsbeauftragten des betreffenden Studienganges. Eine Zulassung kann erteilt werden, wenn absehbar ist, dass die noch offenen Prüfungsleistungen aus dem Grundstudium bis zum Beginn der Praxisphase erbracht werden können.

(2) Die Zulassung zum Praxisprojekt setzt weiterhin die Einreichung folgender Unterlagen an das Praktikantenamt voraus:

- a) Ausgefüllter Antrag auf Zulassung zum Praxisprojekt (Formblatt)
- b) Ausbildungsvertrag (Formblatt der Hochschule oder der Praxisstelle, 3fach),
- c) Ausbildungsplan.

(3) Die unter (2) genannten Unterlagen sind spätestens 4 Wochen vor Beginn der Praxisphase einzureichen. Über Ausnahmen entscheidet der Praktikumsbeauftragte des jeweiligen Studienganges.

(4) Das Praktikantenamt entscheidet aufgrund der eingereichten Unterlagen über die Zulassung

zum Praxisprojekt. Die Zulassung wird auf dem Zulassungsantrag vermerkt.

## **§ 7**

### **Praxisstelle, Betreuung**

- (1) Bei der Auswahl von Praxisstellen werden die Studenten durch den Praktikumsbeauftragten beraten und unterstützt. Jeder Student sollte sich selbst um eine geeignete Praxisstelle und den Abschluss eines entsprechenden Ausbildungsvertrages bemühen. Bleibt die Suche des Studenten erfolglos, so kann ihm eine geeignete Praxisstelle vom Praktikumsbeauftragten zugewiesen werden.
- (2) Mit der Praxisstelle ist ein Ausbildungsplan abzustimmen und schriftlich zu formulieren. Der Ausbildungsplan wird vom Betrieb für die Ausbildung des Studenten entwickelt und ist verbindlich. Er soll die vorgesehenen Tätigkeiten mit den dafür geplanten Zeiten und den Namen der Betreuer im Betrieb enthalten. Der Ausbildungsplan muss den in §§ 3 und 4 genannten Richtlinien für die Ausbildung in der Praxisphase entsprechen.
- (3) Dem für die Informatikstudiengänge zuständigen Praktikantenamt der Fakultät (im Weiteren: Praktikantenamt) obliegt die organisatorische Betreuung des Studenten während der Praxisphase und die Pflege der Beziehungen zu den Praxisstellen. Das Praktikantenamt wird repräsentiert durch die Praktikumsbeauftragten für die Studiengänge Informatik und Medieninformatik.
- (4) Der Student erhält von Seiten der Fakultät einen Hochschullehrer als fachlichen Betreuer, der am Ende auch für die Bewertung des Praxisprojekts verantwortlich ist. Der Student hält Kontakt zum Hochschulbetreuer und unterrichtet ihn regelmäßig über den Fortgang der Arbeiten. Der Student hat das Vorschlagsrecht bei der Auswahl eines Hochschullehrers und kann dabei Unterstützung durch den Praktikumsbeauftragten des jeweiligen Studiengangs erhalten.
- (5) Die Praxisstelle gewährleistet die im Ausbildungsvertrag festgelegten Bedingungen und sichert, dass der Student entsprechend des Ausbildungsplanes eingesetzt wird.
- (6) Während der Praktikantentätigkeit hat der Student die Weisungen des Beauftragten der Praxisstelle zu befolgen und die Arbeitsordnung etc. der Einrichtung einzuhalten.
- (7) Bei Zweifeln am zweckentsprechenden Einsatz des Studenten in der Praxisstelle wirkt der Praktikumsbeauftragte auf Abhilfe hin.
- (8) In Ausnahmefällen, soweit ausreichend Praxisstellen nachweislich nicht zur Verfügung stehen oder ein Praktikum infolge wirtschaftlicher Probleme des Praktikumsbetriebs abgebrochen werden muss, kann das Praxisprojekt durch gleichwertige Teilprojekte ersetzt werden. Die Entscheidung darüber obliegt dem Prüfungsausschuss.

## **§ 8**

### **Ausbildungsvertrag**

- (1) Jeder Student schließt vor Beginn des Praxisprojektes mit der Praxisstelle einen Ausbildungsvertrag ab (Formblätter der Hochschule oder der Praxisstelle).
- (2) Der Ausbildungsvertrag wird in drei gleichlautenden Ausfertigungen von den Vertragsschlie-

ßenden (Student, Praxisstelle) unterzeichnet und vom Praktikumsbeauftragten nach inhaltlicher Prüfung gegengezeichnet. Erst mit dieser Gegenzeichnung ist das Praktikum als Praxisprojekt im Sinne der Studienordnung anerkannt.

(3) Der Student ist während der Praxisphase über die Hochschule unfallversichert. Über alle Gefahren im Betrieb ist der Student in der Praxisstelle zu belehren. Diese Arbeits- und Unfallschutzbelehrung erfolgt aktenkundig zum Tätigkeitsbeginn.

(4) Alle mit dem Ausbildungsvertrag in Verbindung stehenden Ausgaben trägt der Student. Eine Aufwandsvergütung seitens der Praxisstelle ist anzustreben.

(5) Die Hochschule kommt für Schäden, die der Student während der Praxisphase verursacht, nicht auf. Sofern keine Gruppenhaftpflichtversicherung besteht, wird empfohlen, eine private Haftpflichtversicherung für Studierende abzuschließen. Die Praxisstelle ist berechtigt, den Abschluss einer Berufshaftpflichtversicherung zu fordern.

## **§ 9**

### **Anerkennung des Praxisprojektes**

(1) Jeder Student fertigt einen Praktikumsbericht an. Darin sind insbesondere seine Aufgaben während der Praxisphase, die Einbindung seiner Tätigkeit in den Arbeitsablauf der Praxisstelle, Art und Umfang der verwendeten Werkzeuge und Methoden sowie eine persönliche Einschätzung des Nutzeffekts und eventueller Schwierigkeiten im Rahmen des Praxisprojekts wiederzugeben. Der Praxisbericht ist zusammen mit dem von der Praxisstelle zu bestätigenden Tätigkeitsnachweis (Formblatt) beim Praktikantenamt einzureichen.

(2) Praktikumsbericht und Tätigkeitsnachweis sind spätestens zwei Wochen nach Ableistung der Praxisphase im Praktikantenamt abzugeben. Eine Kopie des Praktikumsberichts ist dem betreuenden Hochschullehrer vom Studenten rechtzeitig vor dem Kolloquium zugänglich zu machen.

(3) Zum Praktikumsbericht wird ein Kolloquium durchgeführt. Das Kolloquium wird durch den betreuenden Hochschullehrer bewertet. Auf der Grundlage dieser Bewertung und der im Absatz 2 genannten Unterlagen entscheidet das Praktikantenamt, ob das Praxisprojekt erfolgreich abgeleistet wurde bzw. ob es ganz oder teilweise zu wiederholen ist.

(4) Eine komplette Wiederholung des Praxisprojektes unterliegt den Regelungen für erste und zweite Wiederholungsprüfungen gemäß Prüfungsordnung. Nach einem dritten nicht positiv bewerteten Abschluss des Praxisprojekts hat der Student den Prüfungsanspruch verloren.

(5) Bei unvorhersehbarem und nicht in der Person des Praktikanten begründetem Wechsel der Praxisstelle ist durch Beschluss des Prüfungsausschusses – auch bei geringfügiger Kürzung des Tätigkeitsumfanges – eine Anerkennung des Praxisprojekts möglich.

## **§ 10**

### **Freistellungen**

(1) Während der Praxisphase als festem Studienbestandteil bleibt der Student Mitglied der HTWK Leipzig mit seinen Rechten und Pflichten.

(2) Während der Praxisphase hat der Student keinen Rechtsanspruch auf Urlaub. Die Praxisstelle kann eine Freistellung von bis zu 10 Werktagen gewähren.

(3) Für während der Praxisphase eventuell nachzuholende Prüfungsleistungen ist nach Absprache mit dem Beauftragten der Praxisstelle Freistellung zu gewähren. Der Student muss sich in der Praxisphase für beabsichtigte Prüfungsteilnahmen fristgerecht im Prüfungsamt anmelden.

### **§ 11**

#### **Praxisprojekt im Ausland**

(1) Das Praxisprojekt kann auch in Firmen und Einrichtungen außerhalb Deutschlands absolviert werden, sofern die Tätigkeit den Grundsätzen von § 3 genügt.

(2) Die Rechtsstellung des Studenten ergibt sich auch bei einer Praxisphase im Ausland aus den Bestimmungen von § 8. In Bezug auf Unfall-, Kranken- und Haftpflichtversicherungen sind durch den Studenten die Besonderheiten des Aufenthaltslandes zu berücksichtigen und gegebenenfalls zusätzliche Vorkehrungen zu treffen.

### **§ 12**

#### **Schlussbestimmungen**

Die Anlagen 1-3 (1: Ausbildungsvertrag; 2: Tätigkeitsnachweis; 3: Antrag auf Zulassung) sind verbindliche Formen der Vertragsgestaltung und Berichterstattung.

Anmerkungen:

Anstelle von Anlage 1 kann auch ein von der Praxisstelle vorgegebenes Formular verwendet werden. Ausländische Dokumente sind in deutscher oder englischer Sprache bzw. in beglaubigter Übersetzung vorzulegen.

### **§ 13**

#### **In-Kraft-Treten**

Diese Praktikumsordnung wurde als Bestandteil der StudO vom Fakultätsrat der Fakultät IM beschlossen und gemeinsam mit dieser vom Rektorat genehmigt. Es gelten die Schlussbestimmungen gem. § 6 StudO INB.

## Ausbildungsvertrag

Zwischen der Firma / der Behörde \_\_\_\_\_

Anschrift \_\_\_\_\_

Tel.: (\_\_\_\_\_) \_\_\_\_\_, nachfolgend Praxisstelle genannt, und

und \_\_\_\_\_

Anschrift \_\_\_\_\_

geb. am \_\_\_\_\_ in \_\_\_\_\_

Telefon (\_\_\_\_\_) \_\_\_\_\_, nachfolgend Studentin / Student genannt,

wird nachstehender Vertrag zur Durchführung einer berufspraktischen Tätigkeit (Praxisprojekt) geschlossen, die für das Studium an der

Hochschule für Technik, Wirtschaft und Kultur Leipzig

Fakultät Informatik und Medien

Gustav-Freytag-Straße 42a

04277 Leipzig

im Studiengang \* (Bachelor) vorgeschrieben ist.

### §1 Art und Dauer der Ausbildung

- (1) Die praktische Ausbildung wird in der o.g. Praxisstelle als Praxisprojekt durchgeführt und dauert 12 Wochen.
- (2) Der Vertrag wird für die Zeit vom \_\_\_\_\_ bis \_\_\_\_\_ abgeschlossen.
- (3) Das Praxisprojekt ist Bestandteil des Studiums, der Student bleibt während des Praxisprojektes Mitglied der Hochschule.

### §2 Pflichten der Praxisstelle

Die Praxisstelle verpflichtet sich,

1. die Studentin / den Studenten während des Praxisprojekts entsprechend der Studienordnung, Abschnitt Praxisprojekt, einzusetzen, zu unterweisen und die Durchführung zu überwachen,
2. einen Beauftragten zu benennen, der in allen das Praxisprojekt betreffenden Fragen mit der Hochschule zusammenarbeitet,
3. die Anfertigung des Praktikumsberichtes zu überwachen und diesen zu unterzeichnen,
4. der Hochschule gegebenenfalls von einer vorzeitigen Beendigung des Vertrages oder vom Nichtantritt der praktischen Tätigkeit durch den Studenten Kenntnis zu geben,



5. nach Beendigung der praktischen Tätigkeit der Studentin / dem Studenten schriftlich ein Zeugnis mit Tätigkeitsnachweis auszustellen.

### **§3 Pflichten des Studenten**

Der Student verpflichtet sich,

1. alle ihm gebotenen Ausbildungsmöglichkeiten wahrzunehmen, die im Rahmen seiner Ausbildung übertragenen Arbeiten gewissenhaft auszuführen,
2. die Betriebsordnung und die Unfallverhütungsvorschriften zu beachten sowie Werkzeuge, Geräte und Materialien sorgsam zu behandeln,
3. den Praktikumsbericht sorgfältig anzufertigen und dem Ausbildungsbeauftragten der Praxisstelle vorzulegen,
4. die Interessen der Praxisstelle zu wahren und über Betriebsvorgänge Stillschweigen zu bewahren,
5. bei Fernbleiben die Praxisstelle unverzüglich zu benachrichtigen,
6. bei Erkrankung spätestens am dritten Tag eine ärztliche Bescheinigung vorzulegen.

### **§4 Auflösung des Vertrages**

- (1) Der Vertrag bedarf der Genehmigung der Hochschule. Er verliert seine Gültigkeit, wenn die Voraussetzungen für die Zulassung zum Praxisprojekt gemäß der Studien- und Prüfungsordnung bis zum Vertragsbeginn nicht erfüllt sind.
- (2) Der Vertrag kann von der Praxisstelle gekündigt werden.
  1. aus wichtigen betrieblichen Gründen,
  2. bei Pflichtverletzungen der Studentin / des Studenten.
- (3) Der Vertrag kann durch die Studentin / den Studenten gekündigt werden
  1. bei groben Verstößen gegen den Ausbildungsplan,
  2. wenn sie/er die Ausbildung aus persönlichen Gründen aufgeben möchte.
- (4) Die Kündigung des Vertrages muss schriftlich und unter Angabe der Gründe im Benehmen mit der Hochschule erfolgen.
- (5) Die Genehmigung des Vertrages kann durch die Hochschule aus zwingenden Gründen zurückgezogen werden.

### **§5 Versicherungsschutz**

- (1) Während des Praktischen Studiensemesters ist der Student kraft Gesetzes
  1. nach den Bestimmungen der studentischen Krankenversicherung pflichtversichert,
  2. in der Renten- und Arbeitslosenversicherung beitragsfrei,
  3. gegen Unfall versichert. Im Versicherungsfall übermittelt die Praxisstelle der Hochschule einen Abdruck der Unfallanzeige.

## **§6 Vergütungen**

Die monatliche Vergütung beträgt \_\_\_\_\_ €.

## **§7 Regelung von Streitigkeiten**

Bei allen aus diesem Vertrag entstehenden Streitigkeiten ist vor Inanspruchnahme der Gerichte eine gütliche Einigung unter Mitwirkung der Hochschule anzustreben.

## **§8 Vertragsausfertigung**

Dieser Vertrag wird in drei gleichlautenden Ausfertigungen von der Praxisstelle, dem Studenten und der Hochschule unterzeichnet. Es ist Aufgabe des Studenten, diese Vertragsausfertigungen der Hochschule rechtzeitig vor Vertragsbeginn vorzulegen, und das für die Praxisstelle bestimmte Exemplar dieser wieder zuzuleiten.

## **§9 Sonstige Vereinbarungen**

(ggf. Anlage)

Ort: \_\_\_\_\_

Datum: \_\_\_\_\_

Für die Praxisstelle:

Studentin / Student:

\_\_\_\_\_  
Unterschrift / Stempel

\_\_\_\_\_  
Unterschrift

Von der Praxisstelle wird folgender Beauftragter benannt: \_\_\_\_\_

Dieser Vertrag wird von der Hochschule durch den Praktikumsverantwortlichen der o.g. Fakultät für das Modul „Praxisprojekt“ anerkannt:

Leipzig, den \_\_\_\_\_

\_\_\_\_\_  
Unterschrift/Stempel

# Tätigkeitsnachweis

Herr/Frau \_\_\_\_\_

geb. am \_\_\_\_\_

Studiengang **Informatik Bachelor (INB) / Medieninformatik Bachelor (MIB)**  
**an der Hochschule für Technik, Wirtschaft und Kultur Leipzig**

hat das Praxisprojekt im Zeitraum

vom \_\_\_\_\_ bis \_\_\_\_\_ Wochen

bei \_\_\_\_\_

absolviert.

Kurze Angaben über ausgeübte Tätigkeiten:

---

---

---

---

---

---

---

---

Eintragung von etwaigen Fehlzeiten sowie Information über Freistellungstage:

---

---

---

---

\_\_\_\_\_  
Ort, Datum

\_\_\_\_\_  
Unterschrift und Stempel des Ausbildungsbeauftragten der Praxisstelle

## Antrag auf Zulassung zum Modul Praxisprojekt

Name: \_\_\_\_\_ Vorname: \_\_\_\_\_

Matrikel-Nr.: \_\_\_\_\_ Matrikel: \_\_\_\_\_  
(z.B.: 12INB, 11MIB)

Telefonnummer: \_\_\_\_\_ E-Mail: \_\_\_\_\_

**Ich beantrage die Einwilligung für meine Ausbildung im Praxisprojekt bei der Firma:**

Name: \_\_\_\_\_

Anschrift: \_\_\_\_\_

Telefonnummer: \_\_\_\_\_ E-Mail: \_\_\_\_\_

Betrieblicher Betreuer: \_\_\_\_\_

Betreuender Professor: \_\_\_\_\_  
Name Unterschrift

in der Zeit vom \_\_\_\_\_ bis \_\_\_\_\_

Datum: \_\_\_\_\_ Unterschrift: \_\_\_\_\_  
(Student/in)

### Anlagen

Ausbildungsplan

Ausbildungsvertrag (dreifach)

## Bearbeitungsvermerk

1. Ausbildungsplan:

- akzeptiert
- nicht ausreichend

2. Ausbildungsvertrag:

- akzeptiert
- nicht ausreichend (bei anderer Vertragsvorlage)
- zu beanstanden  
im Punkt \_\_\_\_\_

3. Wiedervorlage:

- zum \_\_\_\_\_

4. Zulassung (vorbehaltlich Paragraph 6 der PraktO-B):

- erteilt am \_\_\_\_\_  
Unterschrift

5. Praxisprojekt wurde erfolgreich abgeschlossen

- Praktikumsbericht eingereicht, Qualität ausreichend
- Tätigkeitsbericht eingereicht (Unterschrift, Stempel)
  
- Vortrag gehalten am \_\_\_\_\_
- Note ( ) im Prüfungsamt eingegangen und registriert am \_\_\_\_\_

Datum: \_\_\_\_\_

Unterschrift: \_\_\_\_\_  
(Praktikumsbeauftragter)