



Hochschule Reutlingen
Reutlingen University



Modulhandbuch Bachelor Medien- und Kommunikationsinformatik (mkiB)

Fakultät Informatik
Hochschule Reutlingen

Studiengang

Bachelor

Medien- und Kommunikationsinformatik (mki)

Stand: 23.01.2019



Inhalt

Modulliste:3

Grafische Darstellung: Curriculum Bachelor Medien- und Kommunikationsinformatik4

Theoretische Grundlagen 15

Theoretische Grundlagen 1 Praktikum7

Informatik 19

Informatik 1 Praktikum.....12

Grafik15

Fotografie18

Theoretische Grundlagen 221

Theoretische Grundlagen 2 Praktikum24

Informatik 227

Informatik 2 Praktikum.....32

Mensch-Maschine-Interaktion.....36

Audio39

Informatik 342

Informatik 3 Praktikum.....46

Softwaretechnik 150

Betriebssysteme.....53

Internetworking.....56

Video60

Datenbanksysteme.....64

Datenbanksysteme Praktikum68

Softwaretechnik 2.....72

Verteilte Systeme76

IT-Sicherheit.....79

Computergrafik.....82

Praktisches Studiensemester85

Wahlpflicht 1 und 2.....88

Mobile Computing90

Recht & BWL.....93

Seminar ausgewählte Themen der Informatik97

Cloud Computing.....100

Mediale Arbeit103

Psychologie.....106

Forschung und Entwicklung109

Bachelor Kolloquium112

Bachelor Thesis.....115

Medien- und Kommunikationsinformatik Projekt 1	118
Medien- und Kommunikationsinformatik Projekt 2	121

Im Folgenden werden die einzelnen Module im Detail beschrieben. Wird nicht anderes erwähnt, sind die zu erbringenden Prüfungsleistungen benotet.

Modulliste:

Semester	Module/Vorlesung	ECTS
1. Semester		
MKIB11	Theoretische Grundlagen 1 Vorlesung	5
MKIB12	Theoretische Grundlagen 1 Praktikum	5
MKIB13	Informatik 1 Vorlesung	5
MKIB14	Informatik 1 Praktikum	5
MKIB15	Grafik	5
MKIB16	Fotografie	5
2. Semester		
MKIB21	Theoretische Grundlagen 2 Vorlesung	5
MKIB22	Theoretische Grundlagen 2 Praktikum	5
MKIB23	Informatik 2 Vorlesung	5
MKIB24	Informatik 2 Praktikum	5
MKIB25	Mensch-Maschine-Interaktion	5
MKIB26	Audio	5
3. Semester		
MKIB31	Informatik 3 Vorlesung	5
MKIB32	Informatik 3 Praktikum	5
MKIB33	Softwaretechnik 1	5
MKIB34	Betriebssysteme	5
MKIB35	Internetworking	5
MKIB36	Video	5
4. Semester		
MKIB41	Datenbanksysteme Vorlesung	5
MKIB42	Datenbanksysteme Praktikum	5
MKIB43	Softwaretechnik 2	5
MKIB44	Verteilte Systeme	5
MKIB45	IT-Sicherheit	5
MKIB46	Computergrafik	5
5. Semester		
MKIB51	Praktisches Studiensemester	30
6. Semester		
MKIB61	Wahlpflicht 1	5
MKIB62	Mobile Computing	5
MKIB63	Recht & BWL	5
MKIB64	Seminar ausgewählte Themen der Informatik	5
MKIB65	Cloud Computing	5
MKIB66	Mediale Arbeit	5
7. Semester		
MKIB71	Wahlpflicht 2	5
MKIB72	Psychologie	5
MKIB73	Forschung und Entwicklung	5
MKIB74	Bachelor-Kolloquium	3
MKIB75	Bachelor-Thesis	12

Grafische Darstellung: Curriculum Bachelor Medien- und Kommunikationsinformatik

- Jedes Modul ist durch einen Block dargestellt, der Modulnamen erscheint links oben.
- Die thematische Zuordnung ergibt sich durch die Farben (siehe Legende).
- Semesterwochenstunden: In jedem Modul ist links unten angegeben, wie viele Semesterwochenstunden (SWS) dem Modul zugeordnet sind.
- Leistungspunkte: Die horizontale Skala unten gibt an, wie viele Leistungspunkte (ECTS) jedem Modul zugeordnet sind.

Medien- und Kommunikationsinformatik																																
Semester		Abschluss Bachelor of Science						Semester																								
7	Wahlfach 4 SWS	Psychologie 4 SWS	Bachelor Kolloquium 2 SWS	Forschung und Entwicklung 2 SWS	Bachelor Thesis			7																								
6	Wahlfach 4 SWS	Seminar Auswahlthemen 2 SWS	Recht und BWL 4 SWS	Mobile Computing 4 SWS	Cloud Computing 4 SWS	Mediale Arbeit 2 SWS	6																									
5	Praxis						5																									
4	Datenbanken Vorlesung 4 SWS	Datenbanken Praktikum 2 SWS	Softwaretechnik 2 4 SWS	Verteilte Systeme 4 SWS	IT-Sicherheit 4 SWS	Computergrafik 4 SWS	4																									
3	Informatik 3 Vorlesung 4 SWS	Informatik 3 Praktikum 2 SWS	Softwaretechnik 1 4 SWS	Betriebssysteme 4 SWS	Internetworking 1 4 SWS	Video 4 SWS	3																									
2	Informatik 2 Vorlesung 4 SWS	Informatik 2 Praktikum 2 SWS	Theor.Grundlagen2 Vorlesung 4 SWS	Theor.Grundlagen2 Praktikum 2 SWS	Mensch Maschine Interaktion 4 SWS	Audio 4 SWS	2																									
1	Informatik 1 Vorlesung 4 SWS	Informatik 1 Praktikum 2 SWS	Theor.Grundlagen1 Vorlesung 4 SWS	Theor Grundlagen1 Praktikum 2 SWS	Grafik 4 SWS	Fotografie 4 SWS	1																									
ECTS		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	ECTS

Thesis
Interdisziplinär
Softwaretechnik
Kommunikation
Medien
Informatik

Modul:	Theoretische Grundlagen 1	
Kürzel:	mkiB11	
Untertitel:		
Lehrveranstaltungen:	Vorlesung	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Dr. Benjamin Himpel	
Dozent(in):	Prof. Dr. Benjamin Himpel	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 1. Semester	
Lehrform/SWS:	Vorlesung	4 SWS
Arbeitsaufwand:	Präsenzstudium	60 Stunden
	Eigenstudium	90 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung :	Mathematik Basiskurs (Rechnen mit Brüchen, Potenzen/Logarithmen, Gleichungen und Gleichungssysteme, Elementare Funktionen)	
Studien-/Prüfungsleistungen/ Prüfungsform:	Vorlesung: Klausur	

Modulziele:

Abstraktion und Modellierung sind wichtige Techniken, um konkrete Probleme besser verstehen und lösen zu können. Ein Modell kann als Formales System aufgefasst werden, der Prozess der Modellbildung ist eine Abstraktion. Das Ziel des Moduls ist es, die Studierenden mit dem Umgang mit Formalen Systemen vertraut zu machen.

Die Grundbausteine Formaler Systeme werden vorgestellt und die Studierenden sollen lernen, wie mit Formalen Systemen gearbeitet wird: Alle Schritte folgen klar definierten Regeln. Der Schwerpunkt der Vorlesung liegt dabei auf diskreten, d.h. endlichen oder abzählbar unendlichen, Strukturen, die in der Informatik besonders häufig vorkommen.

Angestrebte Lernergebnisse:

Kenntnisse:

- Grundbausteine formaler Systeme: Mengen, Relationen und Funktionen.
- Grundbegriffe der Aussagen- und Prädikatenlogik.
- Überblick über den Aufbau des Zahlensystems.
- Zahlendarstellung zu beliebigen Basen.
- Grundbegriffe algebraischer Strukturen: Gruppen, Ringe und Körper.

Fertigkeiten:

- Mengentheoretische Ausdrücke aufstellen, lesen und vereinfachen.
- Beziehungen zwischen Objekten mit Hilfe von Relationen und Funktionen ausdrücken.
- Aussagenlogische und prädikatenlogische Ausdrücke aufstellen, auswerten und vereinfachen.
- Umwandlung von Zahlendarstellungen, Rechnen mit beliebigen Basen.
- Beweise durch Vollständige Induktion und Schleifeninvarianten.
- Rechnen mit Restklassen, Verschlüsseln und Entschlüsseln insbesondere mit dem RSA Verfahren.

Kompetenzen:

- Interpretation von Relationen und Funktionen als Beziehungen zwischen realen Objekten.
- Modellierung von Aussagen mittels Aussagen- und Prädikatenlogik.
- Abstraktion konkreter Operationen wie z.B. Addition und Multiplikation zu Operationen in algebraischen Strukturen.
- Verständnis für Aufzählbarkeit und Iteration.

Inhalt:

Grundlagen von: Aussagen- und Prädikatenlogik, Zahlendarstellung, Mengentheorie, Kombinatorik, Zahlentheorie, algebraischen Strukturen.

Medienformen:

Es findet ein seminaristischer Unterricht mit eingebetteter, gemeinsamer exemplarischer Lösung von Aufgaben und Präsentation von Beispielen an der Tafel statt. Die präsentierten Inhalte werden als Folienskript ausgegeben.

Literatur:

- Schubert, Matthias (2012): Mathematik für Informatiker. Ausführlich erklärt mit vielen Programmbeispielen und Aufgaben. 2., überarb. und erw. Aufl. Wiesbaden: Vieweg + Teubner (Studium).
- Teschl, Gerald; Teschl, Susanne (2013): Mathematik für Informatiker. Band 1: Diskrete Mathematik und Lineare Algebra. 4., überarb. Aufl. 2013. Berlin, Heidelberg: Springer Spektrum (SpringerLink : Bücher).

Modul:	Theoretische Grundlagen 1 Praktikum	
Kürzel:	mkiB12	
Untertitel:		
Lehrveranstaltungen:	Praktikum	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Dr. Benjamin Himpel	
Dozent(in):	Prof. Dr. Benjamin Himpel	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 1. Semester	
Lehrform/SWS:	Praktikum	2 SWS
Arbeitsaufwand:	Präsenzstudium	30 Stunden
	Eigenstudium	120 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	Keine	
Empfohlene Voraussetzung :	Mathematik Basiskurs (Rechnen mit Brüchen, Potenzen/Logarithmen, Gleichungen und Gleichungssysteme, Elementare Funktionen)	
Studien-/Prüfungsleistungen/ Prüfungsform:	Praktikum: Praktikum	

Modulziele

Dieses Modul bietet eine begleitende Übung zur Vorlesung Theoretische Grundlagen 1 (mkiB11) an. Die Studierenden sollen Aufgabenstellungen aus den Bereichen Aussagen- und Prädikatenlogik, Zahlendarstellung, Mengentheorie, Kombinatorik, Zahlentheorie, algebraischen Strukturen verstehen und selbständig bearbeiten können. Hierzu sollen die Studierenden auch mit Brüchen, Potenzen und elementaren Funktionen rechnen sowie Gleichungen und Gleichungssysteme lösen können.

Angestrebte Lernergebnisse:

Kenntnisse:

- Grundbausteine formaler Systeme: Mengen, Relationen und Funktionen.
- Grundbegriffe der Aussagen- und Prädikatenlogik.

- Überblick über den Aufbau des Zahlensystems.
- Zahlendarstellung zu beliebigen Basen.
- Grundbegriffe algebraischer Strukturen: Gruppen, Ringe und Körper.

Fertigkeiten:

- Mengentheoretische Ausdrücke aufstellen, lesen und vereinfachen.
- Beziehungen zwischen Objekten mit Hilfe von Relationen und Funktionen ausdrücken.
- Aussagenlogische und prädikatenlogische Ausdrücke aufstellen, auswerten und vereinfachen.
- Umwandlung von Zahlendarstellungen, Rechnen mit beliebigen Basen.
- Beweise durch Vollständige Induktion und Schleifeninvarianten.
- Rechnen mit Restklassen, Verschlüsseln und Entschlüsseln insbesondere mit dem RSA Verfahren.

Kompetenzen:

- Interpretation von Relationen und Funktionen als Beziehungen zwischen realen Objekten.
- Modellierung von Aussagen mittels Aussagen- und Prädikatenlogik.
- Abstraktion konkreter Operationen wie z.B. Addition und Multiplikation zu Operationen in algebraischen Strukturen.
- Verständnis für Aufzählbarkeit und Iteration.

Inhalt:

Im Praktikum werden Aufgaben aus den Bereichen Aussagen- und Prädikatenlogik, Zahlendarstellung, Mengentheorie, Kombinatorik, Zahlentheorie, algebraischen Strukturen behandelt. Falls notwendig, wird Rechnen mit Brüchen, Potenzen, elementaren Funktionen sowie das Lösen von Gleichungen und Gleichungssystemen geübt.

Medienformen:

Die Studierenden bearbeiten zu Hause individuell oder in Gruppen Übungsaufgaben schriftlich und am PC. Ein Teil der Aufgaben wird durch die Studierenden oder den Dozenten im Praktikum an der Tafel vorgetragen. Offene Fragen und Probleme sollen diskutiert werden.

Material, das in gedruckter und/oder elektronischer Form verteilt wird:

- Übungsaufgaben zum Praktikum.

Literatur:

- Schubert, Matthias (2012): Mathematik für Informatiker. Ausführlich erklärt mit vielen Programmbeispielen und Aufgaben. 2., überarb. und erw. Aufl. Wiesbaden: Vieweg + Teubner (Studium).
- Teschl, Gerald; Teschl, Susanne (2013): Mathematik für Informatiker. Band 1: Diskrete Mathematik und Lineare Algebra. 4., überarb. Aufl. 2013. Berlin, Heidelberg: Springer Spektrum (SpringerLink : Bücher).

Modul:	Informatik 1	
Kürzel:	mkiB13	
Untertitel:	Grundlagen prozeduraler Programmierung	
Lehrveranstaltungen:	Vorlesung	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Dr. Marcus Schöller	
Dozent(in):	Prof. Dr. Marcus Schöller Prof. Dr. Oliver Burgert	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 1. Semester	
Lehrform/SWS:	Vorlesung / Inverted Classroom	4 SWS
Arbeitsaufwand:	Präsenzstudium	60 Stunden
	Eigenstudium	90 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung :	keine	
Studien-/Prüfungsleistungen/ Prüfungsform:	Continuous Assessment (3 Teilklausuren, jede Teilklausur kann noch im Semester wiederholt werden)	

Modulziele:

Die Veranstaltung vermittelt grundlegende Kenntnisse in strukturierter funktionaler Programmierung, Algorithmenentwurf und -bewertung, sowie grundlegender Datenstrukturen. Sie befähigt die Studierenden, kleine und mittlere Problemstellung zu analysieren, geeignete Lösungsalgorithmen zur Lösung von Teilproblemen zu finden, und auf dieser Basis einfache Datenstrukturen und Algorithmen programmiertechnisch umzusetzen. Sie bildet die Basis für alle vertiefenden Informatik-Lehrveranstaltungen. Die Vorlesung wird durch das Praktikum Informatik I ergänzt, in dem die theoretischen Konzepte der Vorlesung praktisch erprobt werden.

Angestrebte Lernergebnisse:

Kenntnisse:

- Methoden zur Beschreibung von Algorithmen und Abläufen (speziell Pseudocode und UML-Aktivitätsdiagramme) kennen.
- Die Schritte der Informationsverarbeitung mittels Computern inkl. Zusammenspiel aus Ein- und Ausgabe, Verarbeitung sowie Speicherung kennen.

- Die grundlegenden Konzepte der strukturierten funktionalen Programmierung kennen. Dazu gehören Verzweigungen, Schleifen, die Arten von Methodenaufrufen sowie das Konzept der Rekursion.
- Typische primitive Datentypen mit deren groben Wertebereichen nennen können.
- Gängige Sortierverfahren von Arrays wie Merge-Sort und Quick-Sort kennen.
- Die Datenstrukturen Listen, Kellerspeicher, sowie Binärbäume und Graphen kennen. Kenntnis von typischen Methoden haben, die auf diese Datenstrukturen angewendet werden.
- Grundlegende Methoden der Laufzeitberechnung (O-Notation) kennen und anwenden können.

Fertigkeiten:

Die Studierenden können die grundlegende Methoden „Teile und Herrsche“, „Induktion“ und „Rekursion“ zum Entwurf von Algorithmen und Programmen anwenden und somit Lösungen für kleinere neue Problemstellungen finden. Dabei nutzen sie die Elemente der strukturierten funktionalen Programmierung und können in „Formale Methoden“ erlernte Techniken der Logik zur Erstellung von Bedingungen in Schleifen und Verzweigungen nutzen. Sie sind in der Lage UML Aktivitätsdiagramme und Pseudocode zur Lösung gegebener Problemstellungen zu erstellen. Die Studierenden sind in der Lage, für Algorithmen eine vereinfachte Laufzeitabschätzung in O-Notation anzugeben und auf dieser Basis unterschiedliche Algorithmen vergleichen. Grundlegende Datenstrukturen und Suchalgorithmen können implementiert werden. Die Studierenden erwerben die Fähigkeit, abstrakte Informatik-Konzepte (Algorithmen und Datenstrukturen) in einer hardwarenäheren Programmiersprache (C) sowie in einer mächtigeren interpretierten Sprache (Python) umzusetzen.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage:

LE#	Lernergebnis (LE)	Geprüft durch
LE1,6, 13, 15	Datentypen und deren Darstellung im Speicher zu verstehen sowie die grundlegenden mechanischen Rechenoperationen nachvollziehen und selbst ausführen zu können.	Klausur
LE2-5	Eine Problemstellung nach dem Teile-und-Herrsche Prinzip zu zerlegen und in Algorithmen zu formalisieren (z.B. als UML Aktivitätsdiagramm).	Klausur
LE7-8	Algorithmen nach den Prinzipien der Induktion zu entwerfen, rekursiv arbeitende Algorithmen zu verstehen und einfache rekursive Algorithmen zu entwickeln.	Klausur
LE9	Gängige Sortierverfahren von Arrays wie Merge-Sort und Quick-Sort erklären zu können.	Klausur
LE10	Die Datenstrukturen Listen, Kellerspeicher, Bäume und Graphen erklären und in einfachen Fällen anwenden zu können.	Klausur
LE 11	Algorithmen mittel O-Notation bewerten können.	Klausur

Inhalt:

Die Vorlesung Informatik I vermittelt grundlegende Konzepte der strukturierten prozeduralen Programmierung sowie Kenntnisse elementarer Datenstrukturen und grundlegender Algorithmen. Zunächst wird die grundlegende Informationsverarbeitung im Rechner vorgestellt

(LE 1), gefolgt von einer Einführung in die Elemente der strukturierten Programmierung (Sequenz, Verzweigung, Schleife) sowie der Algorithmandarstellung in Pseudocode und mittels UML-Aktivitätsdiagrammen (LE 2&3). Die Methodik „Teile-und-Herrsche“ (LE4&5) wird eingeführt, Datenstrukturen wie Felder, Listen, Zeiger, Verbund werden eingeführt (LE 6, 13, 15). Die Entwurfsmethoden Rekursion und Induktion werden eingeführt (LE 7, 8). Gängige Sortierverfahren werden als Beispiele für Algorithmenentwurfsmethoden genutzt (LE 9), Dynamische Datenstrukturen (Listen, Kellerspeicher, Bäume, Graphen) werden vorgestellt und erste Anwendungen dieser Datenstrukturen werden erarbeitet (LE10). Vereinfachte Methoden der Algorithmen-Analyse in O-Notation werden vorgestellt und eingeübt (LE 11).

Medienformen:

Inverted Classroom-Unterricht, bei dem die Studierenden zunächst mit einführenden Aufgaben an den jeweiligen Themenkomplex herangeführt werden, diese einführenden Aufgaben werden in einer Präsenzveranstaltung diskutiert und ggf. vertieft, und im Anschluss werden vertiefende Aufgaben bearbeitet, die dann in einem längeren Unterrichtsblock besprochen und vertieft werden. Die eigenständige Bearbeitung der Aufgaben ist Voraussetzung zur Teilnahme an der Präsenzveranstaltung, wobei auch fehlerhafte eigenständige Lösungen akzeptiert werden.

Den Studierenden stehen Vorlesungsvideos zu jedem Thema sowie Foliensätze und Literaturempfehlungen zur Verfügung, zu jedem Themenblock gibt es Einführungs-, Pflicht- und Vertiefungsaufgaben.

In den Präsenzveranstaltungen kommen Techniken wie Peer-Instruction, Life Programming, Gruppenarbeit, etc. zum Einsatz.

Literatur:

Keine explizite Literaturempfehlung, da nahezu jedes Buch „Algorithmen und Datenstrukturen“ zur Ergänzung der Vorlesungsinhalte geeignet ist. Es werden jedoch in den Vorlesungen regelmäßig Literaturhinweise zur Vertiefung des Stoffs gegeben.

Modul:	Informatik 1 Praktikum	
Kürzel:	mkiB14	
Untertitel:	Grundlagen prozeduraler Programmierung	
Lehrveranstaltungen:	Praktikum	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Dr. Marcus Schölller	
Dozent(in):	Prof. Dr. Marcus Schölller Prof. Dr. Oliver Burgert	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 1. Semester	
Lehrform/SWS:	Praktikum	2 SWS
Arbeitsaufwand:	Präsenzstudium	30 Stunden
	Eigenstudium	120 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung :	keine	
Studien-/Prüfungsleistungen/ Prüfungsform:	Continuous Assessment, mehrere Programmierprüfungen und Projekte, jede Teilprüfung kann einmal im Semester wiederholt werden	

Modulziele:

Die Veranstaltung vermittelt praktische Kenntnisse in strukturierter prozeduraler Programmierung, Algorithmenentwurf sowie grundlegender Datenstrukturen an Hand der hardwarenäheren Programmiersprache C sowie einer höheren interpretierten Sprache (Python), die im Rapid Prototyping Bereich sowie bei der Entwicklung web-zentrierter Anwendungen eingesetzt wird. Die Studierenden lernen, die in der Vorlesung Informatik I erlernten Konzepte der strukturierten funktionalen Programmierung auf eine nach diesen Paradigmen arbeitenden Sprache umzusetzen und sollen in die Lage versetzt werden, selbständig kleinere Programme zur Lösung beschränkter Aufgaben zu erstellen. In dieser Veranstaltung stehen die Grundlagen der prozeduralen und objektorientierten Programmierung (OOP) im Vordergrund. Dabei werden Konzepte aus der Veranstaltung „Theoretische Grundlagen 1“, wie das Rechnen mit verschiedenen Datentypen und der Problematik des Überlaufs sowie Bool'sche Algebra in Bedingungen für Verzweigungen und Schleifen praktisch umgesetzt.

Die Einführung der objektorientierten Programmierung bildet die Grundlage für Veranstaltungen wie „Informatik 2“ und „Informatik 3“, „Datenbanken“ und „Verteilte Systeme“.

In diesem Praktikum werden die zuvor in der Vorlesung „Informatik 1“ erlernten Methoden in Kleingruppen praktisch angewendet und die Einzelleistung in Programmier-Tests abgefragt.

Angestrebte Lernergebnisse:

Kenntnisse:

- Kenntnis ausgewählter Befehle und Programmierkonzepte der Programmiersprachen C und Python.
- Nutzung einer integrierten Entwicklungsumgebung (Eclipse).
- Nutzung von Debugger und Techniken der Fehleranalyse.

Fertigkeiten:

Die Studierenden sind nach Absolvieren des Kurses in der Lage, die Elemente strukturierter Programmierung (Sequenz, Verzweigung, Schleife) zielgerichtet einzusetzen um Fragestellungen zu lösen. Konzepte der Modularisierung von Quelltext sowie Umsetzung algorithmischer Konzepte incl. Rekursion sind bekannt und können selbständig zur Problemlösung eingesetzt werden. Durch Kenntnis unterschiedlicher Techniken der Fehlersuche sind die Studierenden in der Lage, Fehler in Programmen strukturiert zu suchen und zu beheben.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage, vorgegebene Algorithmen mittlerer Komplexität in den Programmiersprachen Python und C umzusetzen. Sie können einfache bis mittelschwere Algorithmen selbst entwerfen und umsetzen, dabei nutzen sie die Techniken der strukturierten Programmierung zielgerichtet. Die gängigen Datenstrukturen inkl. die Arbeit mit Zeigern sind geläufig. Die Studierenden sind in der Lage, Quelltexte zu analysieren und zu verstehen.

Inhalt:

LE 1 dient der Einrichtung der IDE sowie der Einübung der elementaren Kontrollstrukturen „Verzweigung“ und „Schleife“ sowie elementarer Datentypen. LE 2 vermittelt die Nutzung von Feldern und Listen und führt in die Fehlersuche mittels Debugger ein. LE 3 widmet sich dem Thema „Rekursion“, dabei werden eigenständig rekursive Algorithmen programmiert. LE 4 führt weitere Datenstrukturen incl. Zeigern ein. LE 5 beschäftigt sich mit Fehler- und Ausnahmebehandlung, LE 6 mit Dateioperationen. In LE 7 werden dynamische Datenstrukturen (Bäume, Graphen) sowie Sortieren praktisch eingeübt.

Medienformen:

Die Studierenden bereiten sich auf Basis von Videos und Foliensätzen auf die Praktikumsveranstaltung vor und versuchen vorab, Programmieraufgaben zu lösen. Die Veranstaltung besteht aus kurzen Impulsreferaten (~5 Minuten), im Anschluss werden die Konzepte sofort anhand von Einstiegsaufgaben praktisch eingeübt. Dazu steht eine Web-basierte Programmierumgebung zur Verfügung, die den Studierenden umgehend Rückmeldung über ihren Fortschritt und Erfolg bei der Umsetzung der Aufgaben bietet. Um unterschiedlichen Vorkenntnissen und Lerngeschwindigkeiten Rechnung zu tragen, sind die praktischen Aufgaben in der Regel in drei Schwierigkeitsstufen gestaffelt so dass einerseits

sehr fortgeschrittene Studierende eine Herausforderung erhalten, andererseits Studierende mit Einstiegsschwierigkeiten trotzdem über erste Erfolge motiviert werden. Die praktische Programmierfähigkeit wird vom Fachdozenten und mehreren Hilfskräften betreut so dass eine direkte Rückmeldung und Unterstützung ermöglicht wird.

Literatur:

Beliebige Literatur zu C und zu Python.

Modul:	Grafik	
Kürzel:	mkiB15	
Untertitel:		
Lehrveranstaltungen:	Vorlesung Praktikum	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Boris Terpinc	
Dozent(in):	Herr Bernhard Schellmann	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 1. Semester	
Lehrform/SWS:	Vorlesung, Praktikum	4 (2+2) SWS
Arbeitsaufwand:	Präsenzstudium	60 Stunden
	Eigenstudium	90 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung:	keine	
Studien-/Prüfungsleistungen/ Prüfungsform:	Vorlesung: Klausur Praktikum: Projektarbeit	

Modulziele

Ist die Einführung in Theorie und Praxis der Grundlagen der Gestaltung. Im Mittelpunkt steht der Aufbau, Analyse und Einsatz verschiedener Medienbausteine für Druckprodukte und Präsentationen. Der Schwerpunkt liegt in der textlichen, grafischen und bildhaften Gestaltung von Seiten und analysiert gleichzeitig das Zusammenwirken in einem Layout. Dabei werden die zugehörigen Werkzeuge, grafische, fotografische und satztechnische Programme eingesetzt.

Angestrebte Lernergebnisse:

Kenntnisse:

- Kenntnis und Verständnis der Mikro- und Makrotypografie.
- Grundsätze des elementaren Gestaltens kennen und anwenden können.
- Seitengestaltung und Satztechnik kennen und verstehen.
- Prozess des Herstellens von Druckprodukten kennen und selbst umsetzen können.
- Methodik des Erstellens von Präsentationen kennen und können.

- Regelungen und Besonderheiten des Veröffentlichens im Web kennen.
- Gezielter, sinnvoller Einsatz von Farben nachvollziehen können.
- Produktionsplanung durchführen können.

Fertigkeiten:

Die Studierenden lernen das Erstellen von textbasierten Seiten, die Gestaltung von Druck- und Bildschirmseiten mit grafischen Elementen, Farbe und Fotografien. Darüber hinaus wenden Sie gängige grafische, fotografische und Satzprogramme an. Sie stellen ein Druckprodukt aus den einzelnen Medienbausteinen zusammen und gestalten die Inhalte gleichzeitig für eine Bildschirmproduktion. Dabei soll der crossmediale Gedanke in den Mittelpunkt rücken und die Prinzipien des Single-Source-Publishing erkannt werden. In einem kleinen Projektteam werden Kleinprojekte innerhalb des Semesters umgesetzt.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage:

LE#	Lernergebnis (LE)	Geprüft durch
LE1	Elementares Gestalten.	Klausur
LE2	Typografie.	Klausur
LE3	Layout.	Klausur/Artefakt
LE4	Farbe.	Klausur/Artefakt
LE5	Drucktechnische Umsetzung einer Produktion.	Artefakt
LE6	Erstellen einer Präsentation.	Präsentation
LE7	Veröffentlichen im Web.	Klausur

Inhalte

- Mikro- und Makrotypografie.
- Elementares Gestalten.
- Seitengestaltung und Satztechnik.
- Herstellen eines Druckproduktes.
- Erstellen von Präsentationen.
- Veröffentlichen im Web.
- Einsatz von Farben.
- Produktionsplanung.

Medienformen:

Die Vorlesung begleitet das Praktikum. Die Vermittlung des Lehrstoffes wird mit anschaulichen Folien und vielen Beispielen aus dem Elementaren Gestalten erreicht. Die Funktionen der einzelnen Medienbausteine werden mit theoretischen Grundlagen erörtert

aber auch in der Praxis erprobt und getestet. Aktuelle Grafische Programme stehen für das Praktikum und in der Vorlesung als Lehrmittel zur Verfügung.

Literatur:

- Schellmann, Bernhard (2013): Handbuch Medien. Medien verstehen gestalten produzieren. 6., erw. und verb. Aufl. Haan-Gruiten: Verl. Europa-Lehrmittel (Europa-Lehrmittel).

Modul:	Fotografie	
Kürzel:	mkiB16	
Untertitel:		
Lehrveranstaltungen:	Vorlesung Praktikum	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Boris Terpinic	
Dozent(in):	Steffen Schanz	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 1. Semester	
Lehrform/SWS:	Vorlesung, Praktikum	4 (2+2) SWS
Arbeitsaufwand:	Präsenzstudium Eigenstudium	60 Stunden 90 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	Keine	
Empfohlene Voraussetzung:	Keine	
Studien-/Prüfungsleistungen/ Prüfungsform:	Vorlesung: Klausur Praktikum: Projektarbeit	

Modulziele:

Ist die Einführung in Theorie und Praxis der Fotografie. Der Fotoapparat, das Werkzeug des Fotografen und die Bildgestaltung stehen im Fokus. Auf der einen Seite die Funktionsweise analoger und digitaler Fotokameras, die Auflösung, die Formate, das Blitz und Fotolicht. Auf der anderen Seite die Ästhetik und Kreativität des Fotografen, der Bildaufbau und die Aussage. Bildbearbeitung und Bildmanipulation sind weitere Themen.

Angestrebte Lernergebnisse:

Kenntnisse:

- in digitaler Fototechnik und digitalen Bearbeitungsprogrammen.
- der Bildformate und der Bildwandlung.
- im Gestalten von Bildern.
- der verschiedenen Bearbeitungsschritte in der Fotobearbeitung.
- der Farbfotografie und verschiedener Belichtungsvariationen.
- im Einsatz der Beleuchtung und des Blitzlichtes bei Fotoaufnahmen.

Fertigkeiten:

Die Studierenden lernen professionelle Fotokameratechniken kennen, dazu kommen aufnahmetechnische Verfahren mit Blitz- und Studiolicht. Sie bearbeiten eigene Bilder mit digitalen Bildbearbeitungsprogrammen und stellen Bildmaterial für die Weiterverwendung für Druck- oder Webanwendungen her.

Die Studierenden lernen die physikalischen Grundlagen für Licht und Belichtung kennen, geeignete Aufnahmeverfahren auszuwählen und sie lernen professionelle Kameratypen für die Fotoaufnahme anzuwenden. Darüber hinaus lernen sie mit den Bildbearbeitungsprogrammen umzugehen und mit verwandten Gebieten, wie z. B. mit Grafikdesignern und Druckern zusammen zu arbeiten. Sie erwerben die Fähigkeit Bilder in Anwendungen der Informationstechnik, wie in Webauftritten oder Bilddatenbanken, einzusetzen und diese in kleinen Projektteams selbständig auszuführen.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage:

LE#	Lernergebnis (LE)	Geprüft durch
LE1	Phänomene der Lichtausbreitung und Belichtungstechniken zu unterscheiden.	Klausur
LE2	Physikalische und optische Grundlagen und Arten der Bildbearbeitung zu bewerten sowie deren Manipulationsmöglichkeiten anzuwenden.	Klausur
LE3	Bildaufnahmetechniken zu nutzen und Bilder digital zu bearbeiten.	Artefakt
LE4	Bilder richtig zu belichten und sie gestalterisch zu bearbeiten.	Artefakt
LE5	Verarbeitungsformate für Bilder mit verschiedenen technischen Standards zu bewerten und anzuwenden.	Artefakt
LE6	Die erworbenen Kenntnisse eigenverantwortlich in Foto- und Bild Anwendungen umzusetzen und sich im Team zu organisieren.	Artefakt
LE7	Gestalterische Mittel einzusetzen und zu differenzieren.	Artefakt

Inhalt:

Die Studenten erlernen theoretische Grundlagen der Fototechnik. Dabei werden in der Vorlesung Lichtphänomene, Optiken und Aufnahmetechniken behandelt (LE1). Grundlagen fototechnischer Bildbearbeitung, Digitalisierung und Formatierung werden thematisiert (LE2). Die Studenten lernen die Bildaufnahme mit Tageslicht, Kunstlicht und Fotoblitzlicht und die Bearbeitung mit digitalen Programmen. Sie kennen verschiedene Aufnahmetechniken und die Grundprinzipien digitaler Fotokameras (LE3). Bei Fotoaufnahmen im Praktikum werden vielfältige Aufnahmemöglichkeiten thematisiert und ausprobiert. Sie lernen Beispiele aus Werbe- und Objektfotografie kennen die Grundkenntnisse fototechnischer Aufnahmemöglichkeiten (LE4). Sie können mit den wichtigsten Bildformaten umgehen und diese unterscheiden (LE5). Sie organisieren, planen und führen einfache Fotoaufnahmen aus (LE7). Das Praktikum führt in den Umgang mit Aufnahmetechniken und in das Gestalten von Bildern ein. Besondere Formattechniken, wie Panorama- oder 180 Grad Aufnahmen, erweitern herkömmliche Aufnahmetechniken.

Medienformen:

Die Vorlesung begleitet das Praktikum. Die Vermittlung des Lehrstoffes wird mit anschaulichen Folien und vielen Beispielen aus der Fotografie erreicht. Die Funktionen von Kamera, Optik und Licht in der Fotografie wird mit theoretischen Grundlagen erörtert aber auch mit optischen Experimenten vorgeführt. Aktuelle digitale Bearbeitungsprogramme stehen für das Praktikum und in der Vorlesung als Lehrmittel zur Verfügung. Semiprofessionelle Fotokameras und Fotoblitzanlagen sowie ein Filmstudio, das auch als Fotostudio dient, gehören zur technischen Ausrüstung.

Literatur:

- Feininger, Andreas (1977): [Andreas] Feiningers Kompositionskurs der Fotografie. 2. Aufl. Wien: Econ.
- Feininger, Andreas (2003): Die hohe Schule der Fotografie. Das berühmte Standardwerk. Genehmigte Taschenbuchausg., 22. Aufl., 11. Aufl. dieser Ausg. München: Heyne (Heyne-Bücher : 8, Heyne-Ratgeber, 4544 : Heyne-Ratgeber Hobby und Freizeit).
- Vertiefende Literatur wird in der Veranstaltung bekannt gegeben.

Modul:	Theoretische Grundlagen 2	
Kürzel:	mkiB21	
Untertitel:		
Lehrveranstaltungen:	Vorlesung	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Dr. Bernhard Mößner	
Dozent(in):	Prof. Dr. Bernhard Mößner	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 2. Semester	
Lehrform/SWS:	Vorlesung	4 SWS
Arbeitsaufwand:	Präsenzstudium	60 Stunden
	Eigenstudium	90 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung:	mkiB11, mkiB12	
Studien-/Prüfungsleistungen/ Prüfungsform:	Vorlesung: Klausur	

Modulziele:

Kernziel des Moduls ist es, den Begriff der Berechnung anhand verschiedener Modelle vorzustellen. Dazu werden ausgewählte Kapitel der Theoretischen Informatik behandelt: Automatentheorie, Ersetzungssysteme, iterative und rekursive Funktionen bis hin zum Lambda-Kalkül. Neben diesen theoretischen Aspekten werden auch einige für die Praxis relevante Probleme von Berechnungen vorgestellt.

Angestrebte Lernergebnisse:

Kenntnisse:

Aufbau formaler Sprachen, Grammatiken, Automatentheorie; Ersetzung als Berechnungskonzept am Beispiel der Worterzeugung und des Lambda-Kalküls. Zahldarstellung auf dem Computer. Numerik mit Gleitkommazahlen.

Fertigkeiten:

Umwandlung von Grammatiken in Automaten in Sprachen und wieder in Grammatiken soweit wie möglich für die Sprach-Typen der Chomsky Hierarchie. Formalisierung von iterativ und rekursiv gestellten Problemen in formalen Sprachen.

Kompetenzen:

Einsatz von Programmiersprachen zur Formalisierung von Realweltproblemen, Verständnis für Grenzen der praktischen Berechenbarkeit, Reduktion des Problembegriffs auf das Entscheidungsproblem. Den Computer als Maschine verstehen. Erkennen von entscheidbaren/nicht entscheidbaren Problemen. Abstraktion von Problemen. Beurteilung der numerischen Stabilität von Algorithmen.

Inhalt:

Zur Motivation des Begriffs der Berechnung werden Beispiele iterativer und rekursiver Algorithmen vorgestellt: Folge von Heron, Mandelbrotmenge, Nullstellenbestimmung mittels Intervallhalbierung. Dabei ist die Konvergenz von Folgen ein wichtiger Aspekt. Da alle Berechnungen auf einem Computer endlich sind, stellen Lösungen immer nur Näherungen an die exakten Ergebnisse dar.

Die praktischen Probleme bei Berechnungen werden anhand der Zahlendarstellung im Computer und der Gleitkommaarithmetik vorgestellt. Auch hier stellen alle Ergebnisse Näherungen an die exakten Ergebnisse dar, da immer nur mit einer endlichen, fest vorgegebenen Genauigkeit gerechnet werden kann.

Über die Abbildung der Hochsprache in Maschinensprache werden das Ersetzungsprinzip und die Grundprinzipien des Compilerbaus illustriert. Der Syntaxbaum dient dabei als Überleitung in die Grammatiken und damit die formalen Sprachen. Hier werden die Grundlagen der Automatentheorie (Endliche Automaten, Kellerautomat und Turing-Maschine) vermittelt und die Bezüge zwischen Sprachen und Grammatiken innerhalb der Chomsky-Hierarchie vorgestellt. Die Diagonalisierung und das Entscheidungsproblem bilden den Abschluss.

Medienformen:

Vorlesung mit begleitenden Übungen. Seminaristischer Unterricht mit Tafelanschrieb, Tageslichtprojektion und PC-Projektion. Offene Fragen und Probleme sollen diskutiert werden.

Material, das in gedruckter und/oder elektronischer Form verteilt wird:

- Kurzschrift zur Vorlesung.

Literatur:

- Abelson, Harold; Sussman, Gerald Jay; Sussman, Julie (2001): Struktur und Interpretation von Computerprogrammen. Eine Informatik-Einführung. 4., durchges. Aufl. Berlin, Heidelberg: Springer (Springer-Lehrbuch).
- Hämmerlin, Günther; Hoffmann, Karl-Heinz (1991): Numerische Mathematik. 2. Aufl. Berlin: Springer (Grundwissen Mathematik, 7).

- Hollas, Boris (2007): Grundkurs Theoretische Informatik mit Aufgaben und Prüfungsfragen. 1. Aufl. Heidelberg, München: Elsevier Spektrum Akad. Verl (Hochschultaschenbuch).
- Hopcroft, John E.; Ullman, Jeffrey D. (2000): Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie. 4., durchges. Aufl. München, Wien: Oldenbourg.
- Hromkovic, Juraj (2010): Theoretische Informatik. Formale Sprachen, Berechenbarkeit, Komplexitätstheorie, Algorithmik. 04. Aufl. [S.l.]: Vieweg+Teubner Verlag.
- Schöning, Uwe (2008): Theoretische Informatik - kurz gefasst. 5. Aufl. Heidelberg: Spektrum Akademischer Verlag (Hochschultaschenbuch).
- Wegener, Ingo (2005): Theoretische Informatik. Eine algorithmenorientierte Einführung. 3., überarb. Aufl. Wiesbaden: Teubner (Lehrbuch Informatik).

Modul:	Theoretische Grundlagen 2 Praktikum	
Kürzel:	mkiB22	
Untertitel:		
Lehrveranstaltungen:	Praktikum	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Dr. Bernhard Mößner	
Dozent(in):	Prof. Dr. Bernhard Mößner	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 2. Semester	
Lehrform/SWS:	Vorlesung	2 SWS
Arbeitsaufwand:	Präsenzstudium	30 Stunden
	Eigenstudium	120 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung:	mkiB11, mkiB12	
Studien-/Prüfungsleistungen/ Prüfungsform:	Praktikum: Praktikum	

Modulziele:

Dieses Modul bietet eine begleitende Übung zur Vorlesung Theoretische Grundlagen 2 (mkiB21) an. Die Studierenden sollen Aufgabenstellungen aus den Bereichen Automatentheorie, Ersetzungssysteme, iterative und rekursive Funktionen sowie Numerik verstehen und selbständig bearbeiten können.

Angestrebte Lernergebnisse:

Kenntnisse:

Aufbau formaler Sprachen, Grammatiken, Automatentheorie; Ersetzung als Berechnungskonzept am Beispiel der Worterzeugung und des Lambda-Kalküls. Zahldarstellung auf dem Computer. Numerik mit Gleitkommazahlen.

Fertigkeiten:

Umwandlung von Grammatiken in Automaten in Sprachen und wieder in Grammatiken soweit wie möglich für die Sprach-Typen der Chomsky Hierarchie. Formalisierung von iterativ und rekursiv gestellten Problemen in formalen Sprachen.

Kompetenzen:

Einsatz von Programmiersprachen zur Formalisierung von Realweltproblemen, Verständnis für Grenzen der praktischen Berechenbarkeit, Reduktion des Problembegriffs auf das Entscheidungsproblem. Den Computer als Maschine verstehen. Erkennen von entscheidbaren/nicht entscheidbaren Problemen. Abstraktion von Problemen. Beurteilung der numerischen Stabilität von Algorithmen.

Inhalt:

Im Praktikum werden Aufgaben aus den Bereichen iterative und rekursive Folgen, Konvergenz von Folgen, Zahlendarstellung im Computer, Rechnen mit Gleitkommazahlen, Worterzeugung mittels Grammatiken, deterministische und nichtdeterministische endliche Automaten und Turing-Maschinen behandelt.

Medienformen:

Die Studierenden bearbeiten zu Hause individuell oder in Gruppen Übungsaufgaben schriftlich und am PC. Ein Teil der Aufgaben wird durch die Studierenden oder den Dozenten im Praktikum an der Tafel vorgetragen. Offene Fragen und Probleme sollen diskutiert werden.

Material, das in gedruckter und/oder elektronischer Form verteilt wird:

- Übungsaufgaben zum Praktikum.

Literatur:

- Abelson, Harold; Sussman, Gerald Jay; Sussman, Julie (2001): Struktur und Interpretation von Computerprogrammen. Eine Informatik-Einführung. 4., durchges. Aufl. Berlin, Heidelberg: Springer (Springer-Lehrbuch).
- Hämmerlin, Günther; Hoffmann, Karl-Heinz (1991): Numerische Mathematik. 2. Aufl. Berlin: Springer (Grundwissen Mathematik, 7).
- Hollas, Boris (2007): Grundkurs Theoretische Informatik mit Aufgaben und Prüfungsfragen. 1. Aufl. Heidelberg, München: Elsevier Spektrum Akad. Verl (Hochschultaschenbuch).
- Hopcroft, John E.; Ullman, Jeffrey D. (2000): Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie. 4., durchges. Aufl. München, Wien: Oldenbourg.
- Hromkovic, Juraj (2010): Theoretische Informatik. Formale Sprachen, berechenbarkeit, komplexitätstheorie, algorithmik. 04. Aufl. [S.l.]: Vieweg+Teubner Verlag.
- Schöning, Uwe (2008): Theoretische Informatik - kurz gefasst. 5. Aufl. Heidelberg: Spektrum Akademischer Verlag (Hochschultaschenbuch).
- Wegener, Ingo (2005): Theoretische Informatik. Eine algorithmenorientierte Einführung. 3., überarb. Aufl. Wiesbaden: Teubner (Lehrbuch Informatik).

Modul:	Informatik 2	
Kürzel:	mkiB23	
Untertitel:	Erweiterte objektorientierte und prozedurale Programmierung sowie Grundlagen der GUI- und Web-Programmierung	
Lehrveranstaltungen:	Vorlesung	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Dr. Natividad Martinez	
Dozent(in):	Prof. Dr. Natividad Martinez	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 2. Semester	
Lehrform/SWS:	Vorlesung	4 SWS
Arbeitsaufwand:	Präsenzstudium	60 Stunden
	Eigenstudium	90 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung:	mkiB13, mkiB14	
Studien-/Prüfungsleistungen/ Prüfungsform:	Vorlesung: Klausur	

Modulziele:

Aufbauend auf den Grundlagen von „Informatik 1“ werden die Kenntnisse der objektorientierten und prozeduralen Programmierung unter Beibehaltung der UML Notation und der Programmiersprache Java vertieft. Neben der OOP erlangen die Studierenden einen ersten Einblick in die objektorientierte Analyse (OOA), in das objektorientierte Design (OOD) und in die test-getriebene Entwicklung (TDD), um allgemeinere Aufgabenstellungen zu modellieren und anschließend zu implementieren. Zur Datenhaltung werden dieser Veranstaltung die Dateiformate CSV und XML sowie die Java-Serialisierung vorgestellt. Im dritten Semester wird dann in der Datenbanken-Veranstaltung ein größeres Modell zur Datenhaltung vorgestellt.

Während bei der Veranstaltung „Mensch-Maschine Interaktion“ eher die Usability im Vordergrund steht, deckt diese Veranstaltung die Implementierung von Benutzerschnittstellen und Web-Anwendungen im Backend ab. Sie beinhaltet daher eher den technischen Aspekt der Medieninformatik.

Durch den ersten Kontakt der Studenten mit der Web-Programmierung (HTML, HTTP, JSP, Servlets) werden die Grundlagen für Veranstaltungen wie „Internetworking 1“ und „Verteilte Systeme“ gelegt.

Die theoretischen Grundlagen für die tiefere Algorithmik werden in der Veranstaltung „Theoretische Grundlagen 2“ gelegt und in dieser Veranstaltung parallel dazu angewendet.

Die in dieser Vorlesung kennen gelernten Methoden werden im begleitenden Praktikum „Informatik 2“ zeitnah angewendet.

Angestrebte Lernergebnisse:

Kenntnisse:

- Die Phasen einer Software-Entwicklung (GPA, GPM, OOA, OOD, OOP) nennen und klassifizieren zu können.
- Die Notation der UML-Zustandsdiagramme in der Form der Analyse und des Designs, UML-Zustandsdiagramme und Sequenzdiagramme zu kennen.
- Erweiterte Konzepte einer objektorientierten Programmiersprache zu kennen. Dazu gehören die Konzepte von Aggregation & Komposition, innere Klassen, Aufzählungen, Ströme, Ein-/Ausgabe und Serialisierung, Definition, Implementierung und Verwendung von Interfaces.
- Container-Klassen und generische Datentypen zu klassifizieren in Listen, Sets und Maps sowie deren Unterschiede nennen zu können.
- Die Funktionsweise eines Versionierungssystems zu kennen.
- Test-Methoden in Black-, White- und Grey-Box Tests klassifizieren und deren Unterschiede nennen zu können.
- Die komponentenbasierte Struktur von graphischen Benutzeroberflächen zu kennen.
- Verschiedene Methoden der Implementierung von Event-Handling nennen zu können.
- Das Prinzip des Multi-Threadings zu kennen und dessen Probleme (Race Conditions, Deadlocks) nennen zu können.
- Den grundlegenden HTML-Befehlssatz mit HTTP-Kommunikation zwischen einem Client und einem Server zu kennen.
- Die Unterschiede zwischen HTML, XML, Java Server Pages (JSP) und Servlets nennen zu können.
- Die Vorgehensweise eines Backtracking-Algorithmus beschreiben zu können.
- Die Struktur von Graphen zu kennen und die Funktionsweise von einfachen Graphen-Algorithmen (Greedy-Knotenfärbung, Dijkstra-Algorithmus) beschreiben zu können.

Fertigkeiten:

Die Studierenden können die erweiterten Konzepte einer objektorientierten Programmiersprache am Beispiel der Sprache Java implementieren. Sie können einen lesenden und schreibenden Dateizugriff als serialisierten Datenstrom, als Comma Separated Values (CSV) und als XML-Format unter Verwendung von JAXB implementieren. Reguläre Ausdrücke können erstellt werden. Die komponentenbasierte Struktur von graphischen Benutzeroberflächen mit Event-Handling können umgesetzt werden. Zudem können einfache Multi-Threading Anwendungen erstellt werden. Einfache Web-Anwendungen ohne Datenbank-Anbindung, jedoch mit Session-Handling auf der Basis von HTML, JSP und Servlets sind Fähigkeiten, die die Studierenden erlernen. Zusätzlich werden Backtracking-Algorithmen für gegebene Problemstellungen entworfen. Graphen-Datenstrukturen können erstellt und einfache Graphen-Algorithmen verstanden werden.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage:

LE#	Lernergebnis (LE)	Geprüft durch
LE1-6	Die textuelle Aufgabenbeschreibungen formal in UML zu modellieren und als kleinere und mittelgroße Java-Applikation unter Verwendung von erweiterten Konzepten der Objektorientierung wie Interfaces und innere Klassen implementieren zu können. Der Studierende soll dabei auch Testfälle im Vorfeld spezifizieren und Fremd-Bibliotheken bei gegebener JavaDoc-Dokumentation verwenden können.	Klausur
LE7-8	Eine graphische Benutzeroberfläche unter Verwendung von AWT und Swing bei gegebener JavaDoc-Dokumentation in Quellcode erstellen zu können.	Klausur
LE9-11	Ein web-basiertes Frontend unter Verwendung von Java Server Pages (JSPs) und Servlets bei gegebener JavaDoc-Dokumentation zu konzipieren und umsetzen zu können.	Klausur
LE12-13	Backtracking-Algorithmen und Algorithmen zur Graphenfärbung auf gegebene Problemstellungen anwenden und in Java umsetzen zu können.	Klausur

Inhalt:

Diese inhaltlich auf der Veranstaltung „Informatik 1“ aufbauende Veranstaltung vertieft die Kenntnisse der objektorientierten und prozeduralen Programmierung.

Zu der Vertiefung der Objektorientierung gehören die Konzepte von Aggregation & Komposition, innere Klassen (LE1), Aufzählungen, Ströme, Ein-/Ausgabe und Serialisierung sowie Definition, Implementierung und Verwendung von Interfaces [Ratz 2006/07] (LE2),

Es werden verschiedene Arten von Container-Klassen (Listen, Sets, Maps) aus dem Java Collection Framework (JCF) vorgestellt und deren Eigenschaften diskutiert [Ratz 2006/07], Um in einem größeren Anwendungskontext arbeiten zu können, wird die prinzipielle Funktionsweise von Versionierungssystemen am Beispiel von Subversion (SVN) vorgestellt [Pilato 2009]. Dies kann im begleitenden Praktikum optional eingesetzt werden. Um Benutzereingaben in einem solchen Kontext sinnvoll prüfen zu können, werden reguläre Ausdrücke eingeführt [Ullenboom 2011] (LE3).

Im Anschluss daran werden die Unterschiede der UML-Klassendiagramme in der Analyse- und Designphase vorgestellt und zusätzlich Sequenz- und Zustands-Diagramme der UML eingeführt [Balzert 2005]. Dazu gehört auch deren allgemeine Umsetzung in Java-Quellcode (LE4).

Die erweiterte Objektorientierung beinhaltet auch die Einbindung und Anwendung von Fremdbibliotheken in eigene Anwendungen. Unter Berücksichtigung des Studienkontextes der Medien- und Kommunikationsinformatik wird an dieser Stelle die iText-Bibliothek zur dynamischen Erstellung von PDF-Dateien und PDF-Formularen

vorgestellt [Lowagie 2010] sowie die JavaMail-Bibliothek zur automatischen Erstellung und Versand von E-Mails (LE5).

Zusätzlich zu den grundlegenden Testverfahren des Tracings und Debugging aus der Veranstaltung „Informatik 1“ werden in „Informatik 2“ die Unterschiede zwischen Verifikation und Validierung diskutiert und als Test-Methoden der Black-, White- und Grey-Box Tests vorgestellt [Beck 2010] (LE6).

Der zweite größere Teil dieser Veranstaltung ist die Einführung in die komponentenbasierte GUI-Programmierung mit Java AWT und Swing [Zukowski 2004]. Dazu gehören die Steuerelemente, Layout-Manager (LE7) sowie das Event-Handling. Eine blockierende Benutzeroberfläche motiviert das Multi-Threading mit den Problemen der Race Conditions und der Deadlocks [Ullienboom 2011] (LE8).

Der dritte inhaltliche Block dieser Veranstaltung besteht in den Grundlagen der Web-Programmierung. Dazu gehört der grundlegende HTML-Befehlssatz mit HTML-Formularen, die grundlegende Übertragung von Daten zwischen Client und Server mittels HTTP sowie die Verbindung von HTML mit XML und SGML. Zusätzlich wird gezeigt, wie JPG-Bilder (Captchas) mit Java dynamisch erstellt und übertragen werden können (LE9). Die serverseitige Interpretation von HTML-Formularen wird anhand von Java Servlets und Java Server Pages (JSP) vorgenommen [Basham 2008]. In diesem Kontext wird die Verwaltung von HTTP-Sessions und Java-Beans erläutert (LE10). Abschließend wird die Speicherung von Daten im XML-Format [Vonhoegen 2011] vorgestellt sowie die Java Architecture for XML Binding (JaxB) [Scholz, Niedermeier 2009] als Persistenz-Schicht für ganze Objekt-Strukturen (LE11).

Der abschließende vierte Teil dieser Veranstaltung beinhaltet die Vertiefung der prozeduralen Programmierung. Dabei wird die Backtracking-Strategie zur Problemlösung vorgestellt, welches auf dem Konzept der Rekursion aus der Veranstaltung „Informatik 1“ basiert. Als Beispiele sind Sudoku und das n-Damen Problem vorgesehen (LE12). Zusätzlich werden Graphen als Datenstruktur mit Graphenfärbung, Heuristiken (Greedy, Largest-First, Smallest-Last) und Dijkstra-Algorithmus thematisiert [Diestel 2010] (LE13).

Medienformen:

- Seminaristischer Unterricht mit PC-Beamer und Foliensatz sowie Demonstration von Beispielprogrammen und interaktiver Programmentwicklung. Der Foliensatz steht zu Beginn der Veranstaltung zum Download bereit. Teilweise Tafelanschrieb zur Unterstützung der Entwicklung von Algorithmen, Klassen und komplexeren Datenstrukturen.
- Die Vorlesung wird durch E-Learning-Materialien ergänzt, um den Studierenden aus ihrer bisherigen Denkwelt abzuholen und in die formale Denkwelt eines Informatikers einzuführen.

Literatur:

- Balzert, Helmut (2005): Lehrbuch Grundlagen der Informatik. Konzepte und Notationen in UML 2 Java 5 C++ und C# Algorithmik und Software-Technik Anwendungen; mit CD-ROM und e-learning-Online-Kurs. 2. Aufl. Heidelberg: Elsevier Spektrum Akad. Verl (Lehrbücher der Informatik).
- Basham, Bryan; Sierra, Kathy; Bates, Bert (2009): Servlets und JSPs von Kopf bis Fuß. 2. Aufl., dt. Ausg. der 2. engl. Aufl. Beijing, Köln: O'Reilly.
- Beck, Kent (2010): Test-driven development. By example. 15. print. Boston, Mass., Munich: Addison-Wesley (A Kent Beck signature book).
- Diestel, Reinhard (2010): Graphentheorie. 4. Aufl. Heidelberg [u.a.]: Springer (Springer-Lehrbuch Masterclass).

- Lowagie, Bruno (2011): iText in action. 2. ed. Stamford: Manning Publ (Covers iText 5).
- Pilato, C. Michael; Collins-Sussman, Ben; Fitzpatrick, Brian W. (2009): Versionskontrolle mit Subversion. [Software-Projekte intelligent koordinieren]. 3. Aufl., komplett überarb. und aktualisiert. Beijing, Köln: O'Reilly.
- Ratz, Dietmar; Scheffler, Jens; Seese, Detlef (2007): Grundkurs Programmieren in Java. 4., überarb. Aufl. München, Wien: Hanser.

- Ratz, Dietmar (2006): Grundkurs Programmieren in Java. [basiert auf Java 5.0]. 2., aktualisierte u. überarb. Aufl. München, Wien: Hanser
- Scholz, Michael; Niedermeier, Stephan (2009): Java und XML. Grundlagen Einsatz Referenz; [Parse Serialisieren Validieren und Transformieren; alles zu DOM SAX JAXP StAX und JAXB; inkl. Webservices und XML-Publishing mit Cocoon 2.2; CD-ROM mit allen wichtigen Tools und Beispielen]. 2., aktual. und vollst. überarb. Aufl. Bonn: Galileo Press (Galileo Computing).
- Ullenboom, Christian (2012): Java ist auch eine Insel. Das umfassende Handbuch; [Programmieren mit der Java Platform Standard Edition 7; Java von A bis Z: Einführung Praxis Referenz; von Klassen und Objekten zu Datenstrukturen und Algorithmen; aktuell zu Java 7]. 10., aktualisierte und überarb. Aufl. Bonn: Galileo Press (Galileo Computing).
- Vonhoegen, Helmut (2011): Einstieg in XML. [Grundlagen Praxis Referenz; für Anwendungsentwicklung und E-Publishing; Transformation Formatierung Schnittstellen; inkl. XML Schema DTD XSLT CSS XSL XPath DOM SAX SOAP XQuery XForms; nützliche XML-Tools W3C-Dokumente und alle Beispieldaten]. 6., aktualisierte und erw. Aufl. Bonn: Galileo Press (Galileo Computing).
- Zukowski, John (2005): The Definitive Guide to Java Swing. Third Edition. Berkeley, CA: Apress (SpringerLink : Bücher).

Modul:	Informatik 2 Praktikum	
Kürzel:	mkiB24	
Untertitel:	Erweiterte objektorientierte und prozedurale Programmierung sowie Grundlagen der GUI- und Web-Programmierung	
Lehrveranstaltungen:	Praktikum	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Dr. Natividad Martinez	
Dozent(in):	Prof. Dr. Natividad Martinez	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 2. Semester	
Lehrform/SWS:	Praktikum	2 SWS
Arbeitsaufwand:	Präsenzstudium	30 Stunden
	Eigenstudium	120 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung:	mkiB13, mkiB14	
Studien-/Prüfungsleistungen/ Prüfungsform:	Praktikum: Praktikum	

Modulziele:

Aufbauend auf den Grundlagen von „Informatik 1“ werden die Kenntnisse der objektorientierten und prozeduralen Programmierung unter Beibehaltung der UML Notation und der Programmiersprache Java vertieft. Neben der OOP erlangen die Studierenden einen ersten Einblick in die objektorientierte Analyse (OOA), in das objektorientierte Design (OOD) und in die test-getriebene Entwicklung (TDD), um allgemeinere Aufgabenstellungen zu modellieren und anschließend zu implementieren.

Die Daten der zu erstellenden Anwendung werden dieser Veranstaltung noch in Form von Dateien (CSV und XML) gehandhabt, um sie dritten Semester in der Veranstaltungen „Datenbanken“ in ein größeres Datenmodell überführen zu können.

Während bei der Veranstaltung „Mensch-Maschine Interaktion“ eher die Usability im Vordergrund steht, deckt diese Veranstaltung die Implementierung von Benutzerschnittstellen und Web-Anwendungen im Backend ab. Sie beinhaltet daher eher den technischen Aspekt der Medieninformatik.

Durch den ersten Kontakt der Studenten mit der Web-Programmierung (HTML, HTTP, JSP, Servlets) werden die Grundlagen für Veranstaltungen wie „Internetworking 1“ und „Verteilte Systeme“ gelegt.

Die theoretischen Grundlagen für die tiefere Algorithmik werden in der Veranstaltung „Theoretische Grundlagen 2“ gelegt und in dieser Veranstaltung parallel dazu angewendet. In diesem Praktikum werden die zuvor in der Vorlesung „Informatik 2“ erlernten Methoden in Kleingruppen praktisch angewendet und die Einzelleistung in Programmier-Tests abgefragt.

Angestrebte Lernergebnisse:

Kenntnisse:

- Die Notation der UML-Zustandsdiagramme in der Form der Analyse und des Designs, UML-Zustandsdiagramme und Sequenzdiagramme zu kennen.
- Erweiterte Konzepte einer objektorientierten Programmiersprache zu kennen. Dazu gehören die Konzepte von Aggregation & Komposition, innere Klassen, Aufzählungen, Ströme, Ein-/Ausgabe und Serialisierung, Definition, Implementierung und Verwendung von Interfaces.
- Die komponentenbasierte Struktur von graphischen Benutzeroberflächen zu kennen.
- Verschiedene Methoden der Implementierung von Event-Handling nennen zu können.
- Das Prinzip des Multi-Threadings zu kennen und dessen Probleme (Race Conditions, Deadlocks) nennen zu können.
- Den grundlegenden HTML-Befehlssatz mit HTTP-Kommunikation zwischen einem Client und einem Server zu kennen.
- Die Unterschiede zwischen HTML, XML, Java Server Pages (JSP) und Servlets nennen zu können.
- Die Vorgehensweise eines Backtracking-Algorithmus beschreiben zu können.
- Die Struktur von Graphen zu kennen und die Funktionsweise von einfachen Graphen-Algorithmen (Greedy-Knotenfärbung, Dijkstra-Algorithmus) beschreiben zu können.

Fertigkeiten:

Die erweiterten Konzepte einer objektorientierten Programmiersprache am Beispiel der Sprache Java können von den Studierenden implementiert werden. Einen lesenden und schreibenden Dateizugriff als serialisierten Datenstrom, als CSV- und als XML-Format unter Verwendung von JAXB kann implementiert werden. Zusätzlich sind die Lernenden in der Lage reguläre Ausdrücke zu erstellen und programmieren zu können. Fremdbibliotheken wie iText oder JavaMail in bestehende Anwendungen können integriert und genutzt werden. Gray-Box Tests mittels JUnit können implementiert werden. Die komponentenbasierte Struktur von graphischen Benutzeroberflächen mit Event-Handling kann umgesetzt werden. Einfache Multi-Threading Anwendungen können erstellt werden. Außerdem werden Fertigkeiten zur Implementation einfacher Web-Anwendungen ohne Datenbank-Anbindung, jedoch mit Session-Handling auf der Basis von HTML, JSP und Servlets vermittelt. Backtracking-Algorithmen können für gegebene Problemstellungen entworfen und umgesetzt werden. Zusätzlich sind die Studierenden in der Lage, Graphen-Datenstrukturen zu erstellen und einfache Graphen-Algorithmen implementieren zu können.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage:

LE#	Lernergebnis (LE)	Geprüft durch
-----	-------------------	---------------

LE1	Eine größere, aus Kundensicht grob formulierte Aufgabenbeschreibung formal in UML modellieren und als Java-Applikation unter Verwendung von erweiterten Konzepten der Objektorientierung wie Interfaces und innere Klassen implementieren zu können.	Präsentation der Ergebnisse der Kleingruppe sowie durch Programmier-Tests
LE2	Das Backend-Modell zunächst einem Unit-Testing zu unterziehen und anschließend eine graphische Benutzeroberfläche auf ein gegebenes Backend unter Verwendung von AWT und Swing zu konzipieren und umsetzen zu können.	Präsentation der Ergebnisse der Kleingruppe sowie durch Programmier-Tests
LE3	Ein web-basiertes Frontend auf ein gegebenes Backend unter Verwendung von Java Server Pages (JSPs) und Servlets zu konzipieren und umsetzen zu können. Zusätzlich sind die Studierenden in Lage, Algorithmen zur Lösung von komplexeren Problemen (beispielsweise ein Sudoku-Löser mit Backtracking, ein Ressourcen-Scheduling mit Graphenfärbung oder das Finden des kürzesten Weges mit dem Dijkstra-Algorithmus) exemplarisch zu implementieren.	Präsentation der Ergebnisse der Kleingruppe sowie durch Programmier-Tests

Inhalt:

Diese inhaltlich auf der Veranstaltung „Informatik 1“ aufbauende Veranstaltung vertieft die praktische Anwendung der objektorientierten und prozeduralen Programmierung. Dabei werden weitere UML-Diagrammtypen erstellt und in Java-Quellcode umgesetzt.

Im Gegensatz zur Veranstaltung „Informatik 1“ wird in diesem Praktikum jetzt eine einzige komplexere Aufgabe über das gesamte Semester in einer Kleingruppe von 3-4 Studenten bearbeitet. Dazu wird im ersten Teil eine kundenseitige grobe Aufgabenbeschreibung vorgegeben, deren Umsetzung in der Kleingruppe in UML modelliert [Balzert 2005] und als Backend-Applikation zunächst als Konsolen-Anwendung implementiert [Ratz 2006/07], [Ullenboom 2011] wird. Dazu ist das Einbinden und Anwenden von Fremd-Bibliotheken wie iText [Lowagie 2010] und JavaMail erforderlich (LE1).

Im zweiten Teil des Praktikums wird die Java-Applikation als Backend verwendet, um darauf aufbauend eine graphische Benutzeroberfläche unter Verwendung von AWT und Swing [Zukowski 2004] zu implementieren. Zuvor werden erste Unit-Tests des Backends unter Verwendung von JUnit erstellt [Beck 2002] (LE2).

Abschließend wird die Java-Applikation in eine Web-Anwendung unter Verwendung von Java Server Pages (JSP) und Servlets umgewandelt, die von mehreren Benutzern gleichzeitig verwendet werden kann. Die Applikation wird dabei in eine Bean umgewandelt, die von dem Web-Frontend zugegriffen wird (LE3).

Medienformen:

- Die Studierenden bearbeiten in Kleingruppen Übungsaufgaben am PC und entwickeln ihre eigenen Beispielprogramme. Zusätzliche Tests mit Programmieraufgaben zur Prüfung der individuellen Leistung.
- Das Praktikum wird durch E-Learning-Materialien ergänzt, um den Studierenden aus ihrer bisherigen Denkwelt abzuholen und in die formale Denkwelt eines Informatikers einzuführen.
- Intensive Betreuung durch den Dozenten, Assistenten und Tutoren.

Literatur:

- Balzert, Helmut (2005): Lehrbuch Grundlagen der Informatik. Konzepte und Notationen in UML 2 Java 5 C++ und C# Algorithmik und Software-Technik Anwendungen; mit CD-ROM und e-learning-Online-Kurs. 2. Aufl. Heidelberg: Elsevier Spektrum Akad. Verl (Lehrbücher der Informatik).
- Basham, Bryan; Sierra, Kathy; Bates, Bert (2009): Servlets und JSPs von Kopf bis Fuß. 2. Aufl., dt. Ausg. der 2. engl. Aufl. Beijing, Köln: O'Reilly.
- Beck, Kent (2010): Test-driven development. By example. 15. print. Boston, Mass., Munich: Addison-Wesley (A Kent Beck signature book).
- Lowagie, Bruno (2011): iText in action. 2. ed. Stamford: Manning Publ (Covers iText 5).
- Ratz, Dietmar (2006): Grundkurs Programmieren in Java. [basiert auf Java 5.0]. 2., aktualisierte u. überarb. Aufl. München, Wien: Hanser.
- Ratz, Dietmar; Scheffler, Jens; Seese, Detlef (2007): Grundkurs Programmieren in Java. 4., überarb. Aufl. München, Wien: Hanser.
- Ullenboom, Christian (2012): Java ist auch eine Insel. Das umfassende Handbuch; [Programmieren mit der Java Platform Standard Edition 7; Java von A bis Z: Einführung Praxis Referenz; von Klassen und Objekten zu Datenstrukturen und Algorithmen; aktuell zu Java 7]. 10., aktualisierte und überarb. Aufl. Bonn: Galileo Press (Galileo Computing).
- Zukowski, John (2005): The Definitive Guide to Java Swing. Third Edition. Berkeley, CA: Apress (SpringerLink : Bücher).

Modul:	Mensch-Maschine-Interaktion	
Kürzel:	mkiB25	
Untertitel:		
Lehrveranstaltungen:	Vorlesung Praktikum	
Studiensemester:	jedes Semester	
Modulverantwortliche:	Prof. Dr. Gabriela Tullius	
Dozent(in):	Prof. Dr. Gabriela Tullius	
Sprache:	Deutsch, Englisch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 2. Semester	
Lehrform/SWS:	Vorlesung, Praktikum	4 (2+2) SWS
Arbeitsaufwand:	Präsenzstudium:	60 Stunden
	Eigenstudium:	90 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung:	mkiB11, mkiB12, mkiB13, mkiB14, mkiB16	
Studien-/Prüfungsleistungen/ Prüfungsform:	Vorlesung: Klausur Praktikum: Projektarbeit	

Modulziele:

Das Ziel des Moduls ist es, die Studierenden für das Themengebiet Gestaltung von gebrauchstauglichen aber auch attraktiven Benutzungsschnittstellen zu sensibilisieren und sie in die Lage zu versetzen, dementsprechende Lösungen zu entwickeln. Dabei kann das Modul in den ersten beiden Semestern eine zentrale Stellung einnehmen, weil hier mehrere Kompetenzen, die teilweise in anderen Modulen erworben wurden, gebündelt werden können. Im weiteren Verlauf des Studiums soll mit dem erfolgreichen Bestehen gewährleistet werden, dass die Studierenden Anwendungen benutzer-, aufgaben- und kontextgerecht gestalten können.

Angestrebte Lernergebnisse:

Kenntnisse:

- Methoden zur Gestaltung von Benutzungsschnittstellen kennen lernen.
- Elemente des Interaktionsdesignprozesses (gemäß ISO9241) beschreiben können.
- Nutzergruppen und Aufgaben identifizieren und klassifizieren können.

- Richtlinien und Standards für die Gestaltung von Benutzungsschnittstellen benennen und einsetzen können.
- Effizienz, Effektivität, Zufriedenheit erklären können.
- Methoden zur Bewertung von Benutzungsoberflächen kennen.
- Barrierefreiheit, User Experience und Usability erklären können.
- Mobile und Stationäre Benutzungsschnittstellen und deren Kontext kennenlernen
- Unterschiedliche Ein- und Ausgabemodalitäten kennenlernen.

Fertigkeiten:

Die Studierenden analysieren die Nutzergruppe, den Kontext und die Aufgabe der Nutzer. Dazu wenden sie eine Reihe von erlernten Methoden an. Die Studierenden beschreiben Kriterien für die Analyse von Benutzungsschnittstellen. Sie beurteilen Benutzungsschnittstellen nach wissenschaftlichen Kriterien und können grafische Fenstersysteme mit passenden Interaktionsobjekten entwickeln. Die Studierenden konzipieren Anwendungen auch in Hinblick auf Barrierefreiheit (im Kontext von Benutzungsschnittstellen) und können Anwendungen auch diesbezüglich beurteilen. Die Studierenden gestalten Mockups und Prototypen zur sofortigen oder späteren Umsetzung als attraktive und gebrauchstaugliche Produkte. Die Studierenden betrachten dabei das Produkt ganzheitlich unter dem Aspekt der User Experience und wählen je nach Ein- und Ausgabemodalität die passende Gestaltung.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage:

LE#	Lernergebnis (LE)	Geprüft durch
LE1	Unterschiedliche auch abstrakte Vorgehensmodelle für die Entwicklung von Benutzungsoberflächen anzuwenden.	Klausur
LE2	Eine Benutzungsoberfläche nach ergonomischen und ästhetischen Gesichtspunkten für eine gegebene Zielgruppe konzipieren und gestalten zu können.	Projektarbeit
LE3	Die erworbenen Kenntnisse eigenverantwortlich in einem interaktiven Produkt umsetzen zu können.	Klausur, Projektarbeit
LE4	Autorensysteme für die Entwicklung von Benutzungsoberflächen nutzen zu können.	Projektarbeit
LE5	Probleme und Grenzen, die bei der Entwicklung von interaktiven Produkten entstehen, einzuschätzen.	Projektarbeit
LE6	Eigene Entwicklungen und Fähigkeiten beurteilen zu können.	Projektarbeit
LE7	Entwicklungen und Gestaltungsentscheidungen wissenschaftlich fundiert zu kommunizieren, zu präsentieren und zu diskutieren.	Projektarbeit
LE8	Aktuelle Entwicklungen der Disziplin Mensch-Maschine Interaktion zu beurteilen und sich anzueignen.	Projektarbeit

Inhalt:

In der Vorlesung Mensch-Maschine-Interaktion werden die Studenten an die Thematik der Gestaltung von Benutzungsschnittstellen im Sinne eines guten und ergonomischen Interaktionsdesigns herangeführt. Die ganzheitliche Gestaltung der

Benutzungsschnittstelle ist ein wesentliches Element für den Erfolg eines Produkts. So gehört zur Gestaltung der Einsatz von Interaktionsobjekten (Widgets) [Shneiderman 2005] unter kognitiven Aspekten [Johnson 2010] (LE2). Aber auch die Betrachtung und Auseinandersetzung mit dem Nutzerzentrierten Gestaltungsprozess [Sharp et al. 2011] und zugehöriger Methoden [Cooper] (LE1) und der Bewertung der Usability [Sharp et al. 2011] und allgemeiner der User Experience. Die Studierenden setzen sich mit der Frage auseinander, wie Systeme gestaltet werden müssen (LE3, LE5), so dass sie möglichst positiv wirken und nicht nur der pragmatischen, sondern auch hedonischen Qualitäten genügen. Dies bedingt aber auch, dass sich die Studierenden auf unterschiedliche Zielgruppen einstellen und hier ihre kommunikativen und fächerübergreifenden Fähigkeiten anwenden müssen (LE6, LE7).

Im Praktikum üben die Studierenden die Fähigkeit, die in der Vorlesung erworbenen Kenntnisse selbstständig beim Lösen von Übungsaufgaben und bei der Gestaltung von Benutzungsoberflächen anzuwenden. Dazu gehören die Analyse der Zielgruppe und deren Aufgaben sowie der gegebene Kontext. Typischerweise arbeiten die Studierenden dafür an einem größeren Projekt, so dass sie ihr eigenes Vorgehen in einem größeren Bereich reflektieren können. Die Studierenden lernen ein Autorensystem für die Gestaltung von Benutzungsoberflächen kennen und können damit Mockups und Prototypen erstellen (LE4). Die Studierenden recherchieren wissenschaftliche Literatur zum Thema und vertiefen die Vorlesungsinhalte eigenverantwortlich (LE8).

Medienformen:

Das Lehrmaterial besteht aus einem Folienskript, das in elektronischer Form vorliegt, Übungsblättern sowie Einführungen in das Autorensystem. Das Modul umfasst eine Vorlesung mit begleitendem Praktikum. Seminaristischer Unterricht, bei dem Beispiele zu den theoretischen Inhalten multimedial veranschaulicht werden. Die Studierenden bearbeiten individuell oder in Gruppen Übungsaufgaben zum Themengebiet Mensch-Maschine-Interaktion. Die Studierenden entwickeln zunächst mit Papier und Bleistift erste Mockups und dann mit Hilfe eines Autorensystems eigene prototypische Anwendungen. Betreuung durch den Dozenten. Eine umfangreichere Semesterarbeit ist über mehrere Wochen hinweg zu bearbeiten, um die Studierenden an größere Aufgaben heran zu führen.

Literatur:

- Cooper, Alan; Reimann, Robert; Cronin, Dave (2007): About Face 3. The essentials of interaction design. Indianapolis, Ind.: Wiley.
- Johnson, Jeff (2010): Simple guide to understanding user interface design rules. Designing with the mind in mind. Online-Ausg. Burlington, MA: Morgan Kaufmann Publishers.
- Norman, Donald A. (2002): The design of everyday things. 1. Basic paperback ed., [Nachdr.]. [New York]: Basic Books.
- Rogers, Yvonne; Preece, Jenny; Sharp, Helen (2011): Interaction design. Beyond human-computer interaction. 3. ed. Chichester: Wiley.
- Shneiderman, Ben; Plaisant, Catherine (2005): Designing the user interface. Strategies for effective human-computer interaction. 4. ed. Boston, Mass., Munich: Pearson/Addison-Wesley.
- Darüber hinaus aktuelle Artikel aus Fachjournalen und Konferenzen wie bspw. der SIGCHI, i-com oder auch relevante ISO Normen.

Modul:	Audio	
Kürzel:	mkiB26	
Untertitel:		
Lehrveranstaltungen:	Vorlesung Praktikum	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Boris Terpinc	
Dozent(in):	Dipl. Ing. (FH) Jo Baumann	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 2. Semester	
Lehrform/SWS:	Vorlesung, Praktikum	4 (2+2) SWS
Arbeitsaufwand:	Präsenzstudium Eigenstudium	60 Stunden 90 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung:	keine	
Studien-/ Prüfungsleistungen / Prüfungsform:	Vorlesung: Klausur Praktikum: Projektarbeit	

Modulziele:

Das Modul ist die Einführung in Theorie und Praxis der Audioaufnahme, -bearbeitung und Tonstudioteknik. Vermittelt werden grundlegende Kenntnisse akustischer Signale, Signalwege und ihrer Verarbeitung. Phänomene der Schallentstehung, Schallausbreitung und Schallwahrnehmung sind Aspekte und Inhalte sowie Arten von Schallwandlern und Mikrofonen. Das Praktikum führt anhand von kleinen Aufnahmeprojekten in Form von Rundfunkspot-Produktionen, in den Umgang mit Aufnahmetechniken, in Audiosoftware und dem Gestalten von Tondokumenten ein.

Angestrebte Lernergebnisse:

Kenntnisse:

- In Audiotechnik, Mikrofontechnik und der digitalen Schnittprogramme.
- Der Audio - Formate, Wandlung und deren Datentransfer.

- Im Gestalten von Tondokumenten (Radiospots, Kurz-Hörspiele) und der Umsetzung.
- Der verschiedenen Produktionsschritte in der Audioproduktion.
- Der Grundregeln von Mehrspur - Tonstudioaufnahmen.
- Im abmischen verschiedener Tonspuren- und Arten.
- Akustischer Gestaltungsmöglichkeiten.

Fertigkeiten:

Die Studierenden lernen professionelle Tonaufnahme- und Mikrofontechniken kennen, dazu kommen aufnahmetechnische Verfahren. Sie bearbeiten das Tonmaterial mit Audioeditoren und speziellen digitalen Softwareprogrammen, sie mischen und bearbeiten Audiotöne und stellen im sog. Mastering Audiobeiträge her. Die Studierenden lernen die physikalischen und die akustischen Gegebenheiten bei Tonaufnahmen richtig einzuschätzen, geeignete Aufnahmeverfahren und Mikrofontechniken auszuwählen und sie lernen aktuelle professionelle Softwarewerkzeuge zur Audiotbearbeitung anzuwenden. Darüber hinaus lernen sie Töne im Studio aufzunehmen und mit Medienspezialisten aus der Ton- und Bildbranche zusammen zu arbeiten. Sie erwerben die Fähigkeit Tonmedien in Anwendungen der Informationstechnik einzusetzen und in kleinen Projektteams Audioproduktionen selbständig auszuführen.

Angestrebte Lernergebnisse:

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage:

LE#	Lernergebnis (LE)	Geprüft durch
LE1	Phänomene der Schallentstehung und Ausbreitung subjektiv einzuschätzen und zu unterscheiden.	Klausur
LE2	Physikalische und akustische Grundlagen und Signalarten zu bewerten sowie deren Verarbeitungsmöglichkeiten anzuwenden.	Klausur
LE3	Studioschall aufzunehmen und Techniken für Mehrspuraufnahmen zu nutzen und den Tonschnitt handwerklich auszuführen.	Artefakt
LE4	Einen einfachen Rundfunkspot aufzunehmen und abzumischen.	Artefakt
LE5	Aufnahme und Verarbeitungsformate für Bild und Ton mit verschiedenen technischen Standards bewerten und anzuwenden.	Artefakt
LE6	Die erworbenen Kenntnisse eigenverantwortlich in einer Videoproduktion umsetzen und sich als Team zu organisieren.	Artefakt
LE7	Gestalterische Mittel einzusetzen und zu differenzieren.	Artefakt

Inhalt:

Die Studenten erlernen theoretische Grundlagen der Audiotechnik. Dabei werden in der Vorlesung Phänomene der Schallentstehung- und Ausbreitung behandelt (LE1). Grundlagen akustischer Signalbearbeitung, Digitalisierung und Formatierung werden

thematisiert (LE2). Das Anfertigen von Hörbeiträgen und die Umsetzung in ein Audio-Medium Sie bedienen Tonmischpulte und verschiedene Aufnahmetechniken zur Sprach- und Musikaufnahmen (LE3). Während der Tonbearbeitung werden vielfältige Audiobeispiele thematisiert und ausprobiert. Sie kennen die Grundkenntnisse der Tonbearbeitungs- und Mikrofontechniken und der Tonmischung (LE4). Sie können mit den wichtigsten Tonformaten arbeiten und diese unterscheiden (LE5). Sie organisieren, planen und bewerten einfache Tonaufnahmen (LE7). Das Praktikum führt anhand einer Rundfunkspot-Produktion in den Umgang mit der Aufnahmetechnik, der Audiosoftware und dem Gestalten von Tondokumenten ein. Bearbeitung und Mehrspurmischung von Sprach- und/oder Musikclips, Herstellung von Wort- oder Musikbeiträgen als komplettes Medium.

Medienformen:

Die Vorlesung wird unterstützt durch projizierte visuelle Darstellungen, durch Animationen und mit vielen Klangbeispielen. Im Praktikum arbeiten die Studierenden in der Regel in Zweiergruppen selbständig an audiotecnischen Projekten, wie Kurzhörspielen, Rundfunkspots oder Musikclips. Das Lehrmaterial besteht aus Arbeitsunterlagen zur Vorlesung. Die Unterlagen zu Vorlesung und Praktikum liegen in elektronischer Form auf dem Informatik-Server (Relax) des Studiengangs zum Download vor.

Literatur:

- Friedrich, Hans Jörg (2008): Tontechnik für Mediengestalter. Töne hören - Technik verstehen - Medien gestalten. Berlin, Heidelberg: Springer (X.media.press).
- Henle, Hubert (2001): Das Tonstudio-Handbuch. Praktische Einführung in die professionelle Aufnahmetechnik. 5., komplett überarb. Aufl. München: Carstensen (Factfinder-Serie).
- Sandmann, Thomas (2008): Effekte & dynamics. Professionelles Know-How für Mix und Mastering ; die Referenz für Einsteiger und Profis. 7. Aufl. Bergkirchen: PPV Medien.

Modul:	Informatik 3	
Kürzel:	mkiB31	
Untertitel:		
Lehrveranstaltungen:	Vorlesung	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Dr. Marcus Schöller	
Dozent(in):	Dr. Michael Höfling	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 3. Semester	
Lehrform/SWS:	Vorlesung	4 SWS
Arbeitsaufwand:	Präsenzstudium	60 Stunden
	Eigenstudium	90 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung:	mkiB11, mkiB12, mkiB13, mkiB14, mkiB21, mkiB22, mkiB23, mkiB24	
Studien-/Prüfungsleistungen/ Prüfungsform:	Vorlesung: Klausur	

Modulziele

Die Veranstaltung „Informatik 3“ ist die letzte Veranstaltung des Studienplans, bei der das „Programmieren im Kleinen“ im Vordergrund steht, während die Veranstaltungen Softwaretechnik 1 und 2 das „Programmieren im Großen“ behandeln.

Erstes Ziel dieser Veranstaltung ist es zuerst, ein tiefergehendes Verständnis der Methoden der prozeduralen und objektorientierten Programmierung zu vermitteln. Außerdem werden alternative Programmierparadigmen, wie die deklarative, funktionale und aspekt-orientierte Programmierung mit Cross-Cutting Concerns vorgestellt. Um zu zeigen, dass die Konzepte der Programmierung nicht mit einer einzigen Programmiersprache verbunden sind, werden in „Informatik 3“ nun verschiedene andere Sprachen (Eiffel, C/C++, C#, Python) vorgestellt, welche die bereits bekannten Konzepte ebenso umsetzen.

Während die Sprache C# die Studierenden durch ihre Nähe zu Java abholt, wird unter Zuhilfenahme der Sprache C ein tieferes Verständnis für hardwarenahe Programmierung vermittelt. Dies dient der Vorbereitung auf die Veranstaltungen „Computergrafik“ sowie IT-Sicherheit, da „Informatik 3“ bereits auf grundsätzliche Probleme wie Buffer-Overflows und auf Pointer-Probleme eingeht.

Einige bereits bekannte Themen, wie das Konzept der Rekursion, der Nebenläufigkeit und der funktionalen Programmierung werden in „Informatik 3“ vertieft und weiter formalisiert. Dies geschieht bei der Rekursion durch die Vorstellung der verschiedenen Rekursionsarten und bei der Nebenläufigkeit durch die Einführung der Konzepte der Semaphore und Monitore als Vorbereitung zur Veranstaltung „Betriebssysteme“. Bei der funktionalen Programmierung wird vor allem auf das Lambda-Kalkül als mächtiges Werkzeug eingegangen. Darüber hinaus wird auf formaler Ebene der allgemeine Aufbau von Programmiersprachen und deren Repräsentation in EBNF-Notation vorgestellt.

Während die Sprache C# die Studierenden mit bekannten Konzepten durch ihre Nähe zu Java abholt, wird unter Zuhilfenahme der Sprache C/C++ ein tieferes Verständnis für hardwarenahe Programmierung vermittelt, da auf grundsätzliche Probleme wie Buffer-Overflows und auf Pointer eingegangen wird. Die Sprachen Eiffel und Python stehen für Programmiersprachen, die moderne Konzepte wie Spezifikation durch Vertrag bereits seit Jahrzehnten als native Programmierparadigmen unterstützen (Eiffel) oder trotz Skriptsprachen-Charakters nicht mehr aus realen, missionskritischen Softwaresystemen wegzudenken sind (Python).

Die in der Vorlesung kennen gelernten Methoden werden im begleitenden Praktikum „Informatik 3“ zeitnah und praxisnah durch die Verwendung der verschiedenen Sprachen angewendet.

Angestrebte Lernergebnisse:

Kenntnisse:

- Kennen der EBNF-Notation und deren Bedeutung für die Entwicklung von Programmiersprachen.
- Kennen verschiedener aktueller Sprachen, welche die Konzepte der prozeduralen und objektorientierten Programmierung umsetzen.
- Das Pointer-Konzept in C/C++ sowie die Problematik von Buffer-Overflows und Pointer-Probleme kennen und erklären können.
- Kenntnis der Umsetzung von Nebenläufigkeit in verschiedenen Sprachen mit deren Lösungsansätzen bei Problemen, wie Semaphore und Monitore.
- Tiefe Kenntnis des Konzepts der Rekursion.
- Alternative Programmierparadigmen zu der prozeduralen und objektorientierten Programmierung, wie deklarative, funktionale und insbesondere aspekt - orientierte Programmierung zu kennen, beschreiben und bewerten zu können.

Fertigkeiten:

- Schnelles Einarbeiten neuer objekt-orientierter Programmiersprachen durch Anwenden der erlernten OO-Konzepte.
- Aus gegebener EBNF-Notation die Syntax einer Programmiersprache konstruieren und umgekehrt.
- (Kleinere) Programmieraufgaben in verschiedenen Programmiersprachen incl. Nebenläufigkeit implementieren zu können, wobei diese Programmiersprachen für verschiedene Programmierparadigmen ausgelegt sind.
- Verschiedene Rekursionsarten ineinander umwandeln zu können und in iterative Lösungen umprogrammieren zu können.
(Kleine) Parser mit rekursivem Abstieg aus einer gegebenen EBNF-Notation in verschiedenen Programmiersprachen selbst konstruieren können.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage:

Inhalt:

Zu Beginn der Veranstaltung „Informatik 3“ beherrschen die Studierenden bereits die prozedurale und objektorientierte Programmierung aus den Veranstaltungen „Informatik 1 und 2“ unter Verwendung der Programmiersprache Java sowie die wesentlichen Diagrammtypen der UML der objektorientierten Analyse (OOA) und des Designs (OOD). Außerdem werden idealerweise bereits die Theoretischen Grundlagen beherrscht.

Den Einstieg bildet ein Ausblick auf alternative Programmierparadigmen, wie die deklarative Programmierung als Vorbereitung zu SQL in der Datenbank-Veranstaltung, funktionale Programmierung am Beispiel Haskell, F# oder SML sowie in die aspektorientierte Programmierung am Beispiel AspectJ, welche über die Einführung von Cross-Cutting Concerns wieder den Bogen in die objektorientierte Programmierung spannt. Auf diese Weise erhalten die Studierenden verschiedene Sichtweisen auf Ansätze zur Problemlösung. Um zu zeigen, dass die Konzepte der Programmierung nicht mit einer einzigen Programmiersprache verbunden sind, werden neben Java weitere Sprachen (Eiffel, C/C++, C#, Python) vorgestellt, welche die bereits bekannten Konzepte ebenso umsetzen [Block 2011] [Smolka 2011] [Böhm 2005] (LE1).

Ebenso verhält es sich mit dem Thema der Rekursion. Die Vertiefung liegt hier in der Definition der verschiedenen Rekursionsarten sowie in der Umwandlung von rekursiven in iterative Algorithmen [Pomberger, Dobler 2008] (LE2).

Über die Hardwarenähe der Sprache C/C++ werden Studierende an das Pointer-Konzept mit dessen Problemen wie Buffer-Overflows hingeführt und auf die Datendarstellung im realen Speicher fokussiert [Reese 2013] [Klein 2004] (LE3).

Während die Vorlesung das Konzept der Nebenläufigkeit mit deren Problemen wie Deadlocks und Race Conditions und Lösungsansätze wie Semaphore und Monitore eher theoretisch thematisiert, wird die nebenläufige Programmierung anhand der Sprache C# im begleitenden Praktikum zeitnah auch praktisch umgesetzt [Vogt 2012] (LE4).

Nach der Vorstellung der EBNF-Notation und deren Bedeutung wird der Studierende bei seinen bereits bekannten Programmiersprachen abgeholt und die Darstellung dieser Sprachen in die formale EBNF durchgeführt. Hiermit wird die Bedeutung der Meta-Ebene der EBNF verdeutlicht [Pomberger, Dobler 2008] [Balzert 2005] (LE5).

Aufbauend auf der EBNF-Notation erhalten „Informatik 3“-Studierende die theoretischen Grundlagen zum Entwickeln eines eigenen kleinen Parsers im vorlesungsbegleitenden Praktikum und somit einen ersten Einblick in die Grundlagen des Compilerbaus [Harold, Means 2004] (LE6).

Medienformen:

- Seminaristischer Unterricht mit PC-Beamer und Foliensatz sowie Demonstration von Beispielprogrammen und interaktiver Programmentwicklung. Der Foliensatz steht zu Beginn der Veranstaltung zum Download bereit.
- Tafelanschrieb zur Unterstützung.

Literatur:

- Balzert, Helmut (2005): Lehrbuch Grundlagen der Informatik. Konzepte und Notationen in UML 2 Java 5 C++ und C# Algorithmik und Software-Technik

Anwendungen; mit CD-ROM und e-learning-Online-Kurs. 2. Aufl. Heidelberg: Elsevier Spektrum Akad. Verl (Lehrbücher der Informatik).

- Block-Berlitz, Marco; Neumann, Adrian (2011): Haskell-Intensivkurs. Ein kompakter Einstieg in die funktionale Programmierung. Berlin, Heidelberg: Springer Berlin Heidelberg (SpringerLink : Bücher).
- Harold, Elliotte Rusty; Means, W. Scott (2004): XML in a Nutshell. [a desktop quick reference; covers XML 1.1 & XInclude]. 3. ed. Beijing, Köln: O'Reilly.
- Klein, Tobias (2004): Buffer Overflows und Format-String-Schwachstellen. Funktionsweisen Exploits und Gegenmaßnahmen. 1. Aufl. Heidelberg: dpunkt-Verl.
- Pomberger, Gustav; Dobler, Heinz (2008): Algorithmen und Datenstrukturen. Eine systematische Einführung in die Programmierung. München [u.a]: Pearson Studium (it Informatik).
- Reese, Richard (2013): Understanding and using C pointers. [core techniques for memory management]. 1. ed. Beijing, Köln: O'Reilly.
- Smolka, Gert (2011): Programmierung - eine Einführung in die Informatik mit Standard ML. Online-Ausg. München: Oldenbourg.
- Vogt, Carsten (2012): Nebenläufige Programmierung. Ein Arbeitsbuch mit UNIXLinux und Java. München: Hanser Verlag.

Modul:	Informatik 3 Praktikum	
Kürzel:	mkiB32	
Untertitel:		
Lehrveranstaltungen:	Praktikum	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Dr. Marcus Schöller	
Dozent(in):	Herr Daniel O'Grady	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 3. Semester	
Lehrform/SWS:	Praktikum	2 SWS
Arbeitsaufwand:	Präsenzstudium	30 Stunden
	Eigenstudium	120 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung:	mkiB11, mkiB12, mkiB13, mkiB14, mkiB21, mkiB22, mkiB23, mkiB24	
Studien-/Prüfungsleistungen/ Prüfungsform:	Praktikum: unbenotet	

Modulziele:

Erstes Ziel dieser Veranstaltung ist es zuerst, ein tiefergehendes Verständnis der Methoden der prozeduralen und objektorientierten Programmierung zu vermitteln. Um zu zeigen, dass die Konzepte der Programmierung nicht mit einer einzigen Programmiersprache verbunden sind, kommen im Praktikum „Informatik 3“ nun verschiedene andere Sprachen (C++, C#, Eiffel und Python) zur Anwendung, welche die bereits bekannten Konzepte ebenso umsetzen. Auf formaler Ebene wird der allgemeine Aufbau von Programmiersprachen und deren Repräsentation in EBNF-Notation vorgestellt.

Während die Sprache C# die Studierenden durch ihre Nähe zu Java abholt, wird unter Zuhilfenahme der Sprache C++ ein tieferes Verständnis für hardwarenahe Programmierung vermittelt. Dies dient der Vorbereitung auf die Veranstaltungen „Computergrafik“ sowie IT-Sicherheit, da „Informatik 3“ bereits auf grundsätzliche Probleme wie Buffer-Overflows und auf Pointer-Probleme eingeht.

Einige bereits bekannte Themen, wie das Konzept der Rekursion und der Nebenläufigkeit werden im Praktikum „Informatik 3“ vertieft angewendet. Dies geschieht bei der Rekursion

durch die Umsetzung der verschiedenen Rekursionsarten und bei der Nebenläufigkeit durch die Verwendung der Konzepte der Semaphore und Monitore als Vorbereitung zur Veranstaltung „Betriebssysteme“. Zusätzlich dazu werden an das bisherige Wissen anlehrende, neue Themen wie Grundlagen des Compilerbaus, Hashing und ausgeglichene Bäume angewendet, die direkt auf den Inhalten der Veranstaltung „Theoretische Grundlagen 2“ aufbauen. Außerdem werden den Studierenden alternative Programmierparadigmen, wie die deklarative, funktionale und aspektorientierte Programmierung mit cross-cutting concerns anhand von praktischen Aufgabenstellungen näher gebracht.

Angestrebte Lernergebnisse:

Kenntnisse:

- Kennen verschiedener aktueller Sprachen, welche die Konzepte der prozeduralen und objektorientierten Programmierung umsetzen.
- Das Pointer-Konzept in C sowie die Problematik von Buffer-Overflows und Pointer-Probleme kennen und erklären können.
- Tiefere Kenntnis des Konzepts der Rekursion.
- Kenntnis der Umsetzung von Nebenläufigkeit in verschiedenen Sprachen mit deren Lösungsansätzen bei Problemen, wie Semaphore und Monitore.
- Verschiedene Arten des Parsens von Textdateien nennen und erklären zu können.
- Das Konzept des Hashings kennen und erklären zu können.
- Strukturen und Algorithmen zu ausgeglichenen Bäumen zu kennen und erklären zu können.
- Alternative Programmierparadigmen zu der prozeduralen und objektorientierten Programmierung, wie deklarative, funktionale und insbesondere aspektorientierte Programmierung zu kennen, beschreiben und bewerten zu können.

Fertigkeiten:

- (Kleinere) Programmieraufgaben in verschiedenen Programmiersprachen incl. Nebenläufigkeit implementieren zu können, wobei diese Programmiersprachen für verschiedene Programmierparadigmen ausgelegt sind.
- Verschiedene Rekursionsarten ineinander umwandeln zu können und in iterative Lösungen umprogrammieren zu können.
- Kleine DOM- und SAX-Parser selbst in verschiedenen Programmiersprachen konstruieren zu können.
- Problemlösungen unter Verwendung von Hashtables und ausgeglichenen Bäumen erstellen zu können.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage:

LE#	Lernergebnis (LE)	Geprüft durch
LE1	Eigene kleine nebenläufige Programme zu entwickeln und typische Probleme bei der nebenläufigen Programmierung zu identifizieren und lösen zu können. Sie sind in der Lage, typische Algorithmen der ausgeglichenen Bäume für Problemlösungen zu implementieren, das Prinzip des Hashings soweit zu verstehen, dass sie typische Hashing-Algorithmen anwenden und eigene einfache Hashing-Algorithmen	Präsentation der Ergebnisse der Kleingruppe sowie durch Programmier-Tests

	entwickeln und implementieren können. Außerdem sind sie in der Lage, verschiedene Arten des Parsens zu beurteilen sowie eigene kleine Parser zu entwerfen und zu implementieren.	
LE2	Verschiedene Rekursionsarten zu identifizieren, zu vergleichen, zu bewerten und umzuwandeln. Ebenso sind sie in der Lage, rekursive Lösungen in iterative Lösungen umzuwandeln, das Pointer-Konzept der Sprache C anzuwenden und Zugriffe auf Daten im Speicher erklären zu können.	Präsentation der Ergebnisse der Kleingruppe sowie durch Programmier-Tests
LE3	Verschiedene Programmierparadigmen auf konkrete Problemstellungen zu diskutieren und anzuwenden.	Präsentation der Ergebnisse der Kleingruppe sowie durch Programmier-Tests

Inhalt:

Zu Beginn der Veranstaltung „Informatik 3“ beherrschen die Studierenden bereits die prozedurale und objektorientierte Programmierung aus den Veranstaltungen „Informatik 1 und 2“ unter Verwendung der Programmiersprache Java sowie die wesentlichen Diagrammtypen der UML der objektorientierten Analyse (OOA) und des Designs (OOD). Außerdem werden idealerweise bereits die Theoretischen Grundlagen beherrscht.

Ähnlich wie im Praktikum „Informatik 2“ wird in diesem Praktikum eine einzige komplexere Aufgabe, die aus mehreren Komponenten besteht, über das gesamte Semester in einer Kleingruppe von 3-4 Studenten bearbeitet.

Der erste Teil des Praktikums wird durch eine Aufgabenstellung, aufgrund der Nähe zur bereits bekannten Sprache Java, in C# realisiert. Diese Aufgabenstellung enthält nebenläufige Programmierung, so dass die Studierenden mit Problemen wie Deadlocks und Race Conditions konfrontiert werden und diese lösen müssen. Als Datenstrukturen sind unter anderem ausgeglichene Bäume mit deren Algorithmen sowie Hashing-Algorithmen selbst zu implementieren. Die Aufgabenstellung wird so formuliert, dass sowohl die in C# integrierten XML-Parser verwendet werden müssen, als auch eine eigene kleine Parser-Komponente zu einem proprietären Datenformat entwickelt werden muss [Vogt 2012] [Ottmann, Widmayer 2012] [Ottmann, Widmayer 2012] [Harold, Means 2004] (LE1).

Im zweiten Teil des Praktikums erfolgt die Entwicklung in der Sprache C. Hier stehen die verschiedenen Rekursionsarten, die Umwandlung in endrekursive Algorithmen sowie die Umwandlung von rekursiven in iterative Algorithmen im Vordergrund. Eine zweite Teilaufgabe vertieft das Pointer-Konzept der Sprache und den direkten Zugriff auf den Speicher mit der Problematik des Buffer-Overflows [Pomberger, Dobler 2008] [Reese 2013] [Klein 2004] (LE2).

Der letzte Teil des Praktikums bietet einen Einblick in alternative Programmierparadigmen, wie die deklarative Programmierung als Vorbereitung zu SQL in der Datenbank-Veranstaltung, funktionale Programmierung am Beispiel Haskell, F# oder SML sowie in die aspektorientierte Programmierung am Beispiel AspectJ, welche über die Einführung von Cross-Cutting Concerns wieder den Bogen in die objektorientierte Programmierung spannt. Auf diese Weise erhalten die Studierenden verschiedene Sichtweisen auf Ansätze zur Problemlösung [Block 2011] [Smolka 2011] [Böhm 2005] (LE3).

Medienformen:

- Die Studierenden bearbeiten in Kleingruppen Übungsaufgaben am PC und entwickeln ihre eigenen Beispielprogramme.
- Zusätzliche Tests mit Programmieraufgaben zur Prüfung der individuellen Leistung.
- Intensive Betreuung durch den Dozenten, Assistenten und Tutoren.

Literatur:

- Balzert, Helmut (2005): Lehrbuch Grundlagen der Informatik. Konzepte und Notationen in UML 2 Java 5 C++ und C# Algorithmik und Software-Technik Anwendungen; mit CD-ROM und e-learning-Online-Kurs. 2. Aufl. Heidelberg: Elsevier Spektrum Akad. Verl (Lehrbücher der Informatik).
- Block-Berlitz, Marco; Neumann, Adrian (2011): Haskell-Intensivkurs. Ein kompakter Einstieg in die funktionale Programmierung. Berlin, Heidelberg: Springer Berlin Heidelberg (SpringerLink : Bücher).
- Böhm, Oliver (2006): Aspektorientierte Programmierung mit AspectJ 5. Einsteigen in AspectJ und AOP. 1. Aufl. Heidelberg: dpunkt-Verl.
- Harold, Elliotte Rusty; Means, W. Scott (2004): XML in a Nutshell. [a desktop quick reference; covers XML 1.1 & XInclude]. 3. ed. Beijing, Köln: O'Reilly.
- Klein, Tobias (2004): Buffer Overflows und Format-String-Schwachstellen. Funktionsweisen Exploits und Gegenmaßnahmen. 1. Aufl. Heidelberg: dpunkt-Verl.
- Ottmann, Thomas; Widmayer, Peter (2012): Algorithmen und Datenstrukturen. 5. Aufl. Heidelberg: Spektrum Akad. Verl.
- Pomberger, Gustav; Dobler, Heinz (2008): Algorithmen und Datenstrukturen. Eine systematische Einführung in die Programmierung. München [u.a]: Pearson Studium (it Informatik).
- Reese, Richard (2013): Understanding and using C pointers. [core techniques for memory management]. 1. ed. Beijing, Köln: O'Reilly.
- Smolka, Gert (2011): Programmierung - eine Einführung in die Informatik mit Standard ML. Online-Ausg. München: Oldenbourg.
- Vogt, Carsten (2012): Nebenläufige Programmierung. Ein Arbeitsbuch mit UNIXLinux und Java. München: Hanser Verlag.

Modul:	Softwaretechnik 1	
Kürzel:	mkiB33	
Untertitel:		
Lehrveranstaltungen:	Vorlesung Praktikum	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Dr. Christian Kücherer	
Dozent(in):	Prof. Dr. Christian Kücherer	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 3. Semester	
Lehrform/SWS:	Vorlesung, Praktikum	4 (2+2) SWS
Arbeitsaufwand:	Präsenzstudium	60 Stunden
	Eigenstudium	90 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung:	mkiB11, mkiB12, mkiB13, mkiB14, mkiB21, mkiB22, mkiB23, mkiB24,	
Studien-/Prüfungsleistungen/ Prüfungsform:	Vorlesung: Klausur Praktikum: Praktikum	

Modulziele

Die Studierenden sollen das Wissensgebiet Softwaretechnik kennenlernen. Basis- und Modellierungskonzepte aus den Grundlagenveranstaltungen werden im Kontext der Entwicklung größerer Softwaresysteme zur systemischen Modellierung im Umfeld der Softwareentwicklung weiterentwickelt. Die Studierenden erlangen außerdem ein grundsätzliches Verständnis der Methoden und Abläufe bei der Entwicklung großer Softwaresysteme im Hinblick auf die weiterführenden Veranstaltungen des Studienganges. Weiterhin werden die zur Softwaretechnik gehörende Techniken im Qualitätsmanagement, dem Projektmanagement der Entwicklung und Evolution im Softwareentwicklungsprozess als Mehrwert den Studierenden zugänglich gemacht.

Angestrebte Lernergebnisse:

Kenntnisse:

Die Studierenden kennen die essentiellen Basiskonzepte und Modellierungsprinzipien und die begrifflichen Grundlagen der Softwaretechnik. Sie kennen die wesentlichen Merkmale von traditionellen und agilen Methoden und Prozessen in der Softwareentwicklung, die für die Entwicklung größerer und großer Systeme eingesetzt werden. Sie stellen fest, dass es bei aufwendigen Softwareprojekten, bei traditionellen aber auch bei agilen Methoden eine Notwendigkeit für ein spezifisches Softwaremanagement gibt und damit Entwicklungs- und Aufgaben- und Produktkomplexität beherrschbar, planbar und Projekte wirtschaftlich realisierbar gemacht werden. Im Einzelnen werden die nachfolgenden Kenntnisse vermittelt, zu Basiskonzepten und zur Modellierung:

- Grundlagen der Struktur- und Verhaltensmodelle softwarebasierter Systeme.

zu Methoden und Abläufen der Softwareentwicklung:

- Traditionelle und agile Entwicklungsprozesse,
- Anforderungsmanagement,
- Entwurf und Entwicklung,
- Verifikation und Validierung.

und zu Managementaktivitäten in der Softwareentwicklung:

- Projektmanagement,
- Qualitätsmanagement,
- Konfigurationsmanagement,
- Life Cycle Management.

Fertigkeiten:

In den Übungen konstruieren die Studierenden Verhaltens-, Struktur- und Datenmodelle und wenden die Kenntnisse über Basiskonzepte und Modellierung aus dieser und den Grundlagenveranstaltungen im softwaretechnisch relevanten Kontext an. Sie wenden die in der Vorlesung besprochenen Techniken selbst auf eine Problemstellung an, um tiefere Kenntnis der Softwareentwicklung unter Berücksichtigung von Qualitätsaspekten für große Systeme zu erlangen.

Nach etablierten Methoden der Softwaretechnik bestimmen sie im Team Anforderungen und spezifizieren Test- und Anwendungsfälle, sowie User Stories. Sie bewerten Anforderungen und Testfälle.

Kompetenzen:

LE#	Lernergebnis (LE)	Geprüft durch
LE1	Die kompetente Verwendung der Fachsprache der Softwaretechnik. Nachweis methodischer Kompetenzen in der Softwaretechnik, speziell in den Basiskonzepten, der Modellierung, den Methoden, den Abläufen und in den Managementaktivitäten in der Softwareentwicklung.	Klausur
LE2	Anwendungskompetenz zu den Methoden der Softwaretechnik, speziell in den Basiskonzepten, der Modellierung, den Methoden und den Abläufen im Softwaremanagement und in der Softwareentwicklung.	Praktikum

Inhalt:

Vorlesung und Übung gliedern sich konsistent in die vier Teilgebiete, in denen die Kenntnisse und Fähigkeiten vermittelt werden, um die Kompetenzen zu erreichen und die Modulziele zu verwirklichen:

- Basiskonzepte und Modellierung in der Entwicklung,
- Qualitätsmanagement und Abläufe der Softwareentwicklung,
- Projektmanagement in der Softwaretechnik,
- Evolution von Softwaresystemen.

Maßgeblich werden Basiskonzepte der Modellierung und Managementmethoden weitgehend den Lehrbüchern von Tabeling, Ludewig & Lichter, Sommerville und Balzert nachempfunden. In den Abläufen der Softwareentwicklung spielen die praxiskonformen Darstellungen insbesondere bei den Übungen nach Rupp eine wichtige Rolle.

Medienformen:

Themenabhängig werden seminaristische und klassische Vorlesungsformen mit Tafelanschrieb und Präsentationsfolien über PC-Beamer verwendet. Die Vorlesungsfolien werden über die Plattformen der Hochschule den Studierenden zugänglich gemacht.

Literatur:

- Ludewig, Jochen; Lichter, Horst (2013): Software Engineering. Grundlagen, Menschen, Prozesse, Techniken. 3. Aufl. Heidelberg: Dpunkt-Verlag
- Balzert, Helmut (2011): Lehrbuch der Softwaretechnik Entwurf, Implementierung, Installation und Betrieb. 3. Aufl. Heidelberg: Spektrum Akademischer Verlag (SpringerLink : Bücher).
- Rupp, Chris; SOPHISTen, die (2014): Requirements-Engineering und -Management. Aus der Praxis von klassisch bis agil. 6., aktualisierte und erweiterte Auflage. München: Hanser, Carl.Balzert (hrsg): Lehrbuch der Softwaretechnik
- Tabeling, Peter (2006): Softwaresysteme und ihre Modellierung. Grundlagen, Methoden und Techniken ; mit 45 Tabellen. Berlin [u.a.]: Springer (EXamen.press).
- Sommerville, Ian (2012): Software Engineering. 9., aktualisierte Aufl. München: Pearson (Pearson Studium - IT).

Modul:	Betriebssysteme	
Kürzel:	mkiB34	
Untertitel:		
Lehrveranstaltungen:	Vorlesung Praktikum	
Studiensemester:	jedes Semester	
Modulverantwortliche:	Prof. Dr. Christian Kücherer	
Dozent(in):	Prof. Dr. Christian Kücherer	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 3. Semester	
Lehrform/SWS:	Vorlesung, Praktikum	4 (2+2) SWS
Arbeitsaufwand:	Präsenzstudium	60 Stunden
	Eigenstudium	90 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:		
Empfohlene Voraussetzung	mkiB11, MkiB12, mkiB13, mkiB14, mkiB21, mkiB22, mkiB23, mkiB24	
Studien-/Prüfungsleistungen/ Prüfungsform:	Vorlesung: Klausur Praktikum: Praktikum	

Modulziele

Das Ziel dieses Moduls ist, Aufgaben, Grundbegriffe, Strukturen und Konzepte von Betriebssystemen anhand verbreiteter Betriebssystemfamilien kennenzulernen. Die Studierenden werden einen tiefergehenden Einblick in die Rechnertechnik gewinnen. Dabei lernen Studierende die Vorgehensweisen zur Lösung von Problemen beim Erstellen von systemnaher Software. Diese Vorlesung bietet Grundlagen für die spätere Vorlesungen Verteilte Systeme, IT-Sicherheit und Mobile Computing, u.a.

Angestrebte Lernergebnisse:

Kenntnisse:

- Die Studierenden kennen die Organisation eines Rechnersystems.

- Die Studierenden kennen die Struktur eines Betriebssystems und klassifizieren dessen Services.
- Die Studierenden kennen die Grundlagen der unterschiedlichen Gebiete eines Betriebssystems: Prozessverwaltung, virtuelle Speicherverwaltung, Dateiverwaltung und persistente Speicherverwaltung.
- Die Studierenden kennen die Prinzipien der Nebenläufigkeit und der Synchronisation zwischen Prozessen und Threads.

Fertigkeiten:

- Die Studierenden arbeiten sich in unbekannte Softwareumgebungen ein und gehen mit verschiedenen Betriebssystemen um.
- Die Studierenden wenden CPU-Scheduling und Speicher-Allocation Algorithmen an.
- Die Studierenden programmieren die Lösung zu Nebenläufigkeitsproblemen mit Hilfe von Synchronisationsmechanismen.
- Die Studierenden dimensionieren Systeme zur Verwaltung von Speicher.
- Die Studierenden benutzen die grundlegenden betriebssystemnahen Befehle und können diese mit Skripten programmieren.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage:

LE#	Lernergebnis (LE)	Geprüft durch
LE1	Die Abläufe im Hintergrund beim Benutzen eines Rechners erklären und analysieren zu können.	Klausur
LE2	Die passenden Algorithmen zu Ressourcenverwaltung (CPU, Speicher, Disc) zu wählen und anzuwenden.	Klausur, Praktikum
LE3	Systemnahe Software zu konzipieren und zu entwickeln.	Klausur, Praktikum
LE4	Im Team zu arbeiten, um komplexe Aufgaben zu lösen.	Praktikum
LE5	Betriebssysteme zu installieren, zu konfigurieren und zu verwalten.	Praktikum
LE6	Mit Betriebssystem-Spezialisten zielführend kommunizieren zu können.	Praktikum

Inhalt:

Die Inhalte der Vorlesung werden in mehrere Themen aufgeteilt. Zu jedem Thema werden die Aufgaben des Betriebssystems (LE1) und die Algorithmen für die Aufgabenlösungen (LE2) erklärt. Die Studierenden gewinnen Fertigkeiten durch praktische Anwendung der Algorithmen in Übungen oder in Form von systemnaher Software (LE3), die teilweise im Labor entwickelt und getestet wird. Bei den praktischen Tätigkeiten werden die Studierenden in Teams arbeiten (LE4). Die Hauptreferenz ist [Tannenbaum] und [Silberschatz].

In der Vorlesung werden folgende Inhalte bearbeitet:

- Grundlagen, allgemeine Konzepte wie Prozess, Speicher, Ein-/Ausgabe, Datei, Architekturmodelle wie Schichten, Kern, μ -Kern, Objektorientierung und Kommunikationsmodelle; Aufgaben, Konzepte, Strukturen und Architekturen von lokalen, Netzwerk- und verteilten Betriebssystemen; Hardware-Software-Schnittstelle (CPU, Motherboard, Peripherie).

- Prozessverwaltung, Nebenläufigkeit und Kooperation sequenzieller und verteilter Prozesse, Koordinations-, Synchronisations- und Kommunikationsprobleme und –mechanismen,.
- Reale und virtuelle Speicherverwaltung.
- Organisation von persistentem Speicher.
- Dateiverwaltung.
- Spezielle Aspekte wie Administration, Sicherheit, Fehlertoleranz.
- Betriebssysteme für verschiedene Einsatzbereiche wie Rechnernetze (Netzwerkbetriebssysteme, verteilte Betriebssysteme), eingebettete Systeme (z.B. Echtzeitbetriebssysteme) und mobile Systeme.
- Konkrete Betriebssysteme wie Unix, Linux, Windows (16- und 32-Bit) werden ausgewählt, untersucht und hinsichtlich ihrer Entwicklung, ihrer Konzepte, Strukturen und Architekturen, ihrer Rollen in der Betriebssystem-Welt und der Möglichkeiten und Grenzen ihres Einsatzes verglichen (LE5, LE6).

Medienformen:

Die Vorlesung folgt einem seminaristischen Stil. Basierend auf Folien-Skripten werden die Konzepte, Strukturen und Algorithmen auf der Tafel veranschaulicht und mit integrierten Übungen angewandt. Passend zu den Skripten wird auf die Kapitel des Textbuches verwiesen. Das Material wird im Voraus in der eLearning-Plattform zur Verfügung gestellt. Darüber hinaus werden Aufgaben im Labor gelöst, dafür werden Einleitungen in Form von Tutorien und programmierbaren Übungen vorbereitet.

Literatur:

- Tanenbaum, Andrew S. (2009): Modern operating systems. 3. ed., Pearson internat. Ed. Upper Saddle River, NJ: Pearson Prentice-Hall.
- Stallings, William (2003): Betriebssysteme. Funktion und Design. Pearson Studium - IT-
- Silberschatz, Abraham; Galvin, Peter B.; Gagne, Greg (2010): Operating system concepts with Java. 8th ed. Hoboken, NJ: John Wiley & Sons.

Modul:	Internetworking	
Kürzel:	mkiB35	
Untertitel:	Netze und Protokolle	
Lehrveranstaltungen:	Vorlesung Praktikum	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Dr. Marcus Schöller	
Dozent(in):	Prof. Dr. Marcus Schöller	
Sprache:	Deutsch, Englisch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 3. Semester	
Lehrform/SWS:	Vorlesung, Praktikum	4 (2+2) SWS
Arbeitsaufwand:	Präsenzstudium	60 Stunden
	Eigenstudium	90 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung:	keine	
Studien-/Prüfungsleistungen/ Prüfungsform:	Vorlesung: Klausur Praktikum: Praktikum	

Modulziele:

Diese Veranstaltung vermittelt Kenntnisse sowohl über die Infrastruktur, als auch über die Kommunikationsprotokolle des Internets. Es werden die wesentlichen Internet-Protokolle von der Netzzugangsschicht bis zur Anwendungsschicht besprochen und in das ISO/OSI-Referenzmodell eingeordnet. Im praktischen Teil der Veranstaltung werden zentrale Konzepte dieser Protokolle selbst umgesetzt und analysiert.

In dieser Veranstaltung wird das Basiswissen für aufbauende Veranstaltungen wie „Verteilte Systeme“ und „IT-Sicherheit“ im folgenden 4. Semester, sowie für „Mobile Computing“ und „Cloud Computing“ im 6. Semester des Studiengangs aufgebaut.

Angestrebte Lernergebnisse:

Kenntnisse:

- Die Schichten und Aufgaben des ISO/OSI-Referenzmodells sowie des TCP/IP/IP-Modells nennen zu können.
- Typische Internet-Protokolle (wie HTTP, FTP, SMTP, DNS, TCP, UDP, IP, ICMP, DHCP, ARP, CSMA/CD, CSMA/CA, TDMA, Ethernet und WLAN) zu nennen, deren

Funktionsweise und Protokoll-Header beschreiben zu können und die Protokolle in die Schichten der Referenzmodelle einordnen zu können.

- Den Stop-and-Wait-Algorithmus und den Sliding-Window-Algorithmus erklären zu können.
- Methoden der Überlastkontrolle in TCP beschreiben zu können.
- Verschiedene Arten des Framings (Byte-Count Methode, Sentinel-Methode) zu kennen.
- Grundlegende Begriffe und Verfahren der Fehlererkennung (Hamming-Distanz, Paritäten, CRC) nennen und beschreiben zu können.
- Leistungsparameter für Netze zu kennen.
- Typische Vermittlungsarten, Netztopologien und deren Eigenschaften nennen zu können. Die entsprechende Hardware, die am Aufbau einer Internet-Infrastruktur beteiligt ist (wie Router, Switches und Hubs) benennen und deren Funktionsweise beschreiben zu können.
- Das Hidden-Station-Problem und Lösungsansätze (MACA und MACAW) erklären zu können.

Fertigkeiten:

- Anbindungen von Heim- und Intranet-Rechnern an das Internet konstruieren zu können.
- Socket-Verbindungen und einfache Client-Server Anwendungen in Java programmieren zu können.
- Beispiele des Distance Vector Routings, des Link State Routings und des Spanning-Tree Algorithmus anhand eines Graphen rechnen zu können.
- Internet-Kommunikation mit Hilfe eines Sniffers zu protokollieren und auswerten zu können.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage:

LE#	Lernergebnis (LE)	Geprüft durch
LE1	Bekannte und neue Protokolle und Netzwerk-Technologien zu analysieren und in das ISO/OSI- und in das TCP/IP-Referenzmodell einzuordnen. Dazu sind die Studierenden in der Lage, Datenpakete zu erfassen und zu interpretieren. Die Aufgaben von typischen Protokollen der Anwendungsebene soweit zu verstehen, dass sie die Funktionsweise von weiteren Protokollen einordnen und beurteilen können. Sie können einfache Infrastrukturen entwerfen und beurteilen.	Klausur
LE2	Zu beurteilen, für welche Anwendungen sich eher TCP oder eher UDP eignet. Dazu können sie die Eigenschaften und Algorithmen der Protokolle diskutieren, außerdem können sie eigene kleine Anwendungen über Socket-Programmierung implementieren.	Klausur
LE3	Typische Protokolle der Vermittlungsschicht mit deren Aufgaben zu verstehen und deren Anwendungsgebiete zu beurteilen. Sie können Subnetze konzipieren und berechnen. Zusätzlich setzen sie Router ein und wenden	Klausur

	verschiedene Routing-Protokolle anhand von Beispielen an.	
LE4	Typische Verfahren und Parameter zu kennen und anzuwenden, um Nutzdaten mit Frames zu umhüllen. Sie diskutieren den Sinn des Framings und beschreiben die zugehörige Hardware.	Klausur
LE5	Den Übergang von Daten(frames) in elektronische Signale und umgekehrt zu diskutieren und gängige Vermittlungsarten und Netztopologien mit deren Eigenschaften und Problemen zu diskutieren.	Klausur

Inhalt:

Diese Veranstaltung holt den Studenten von den Veranstaltungen „Informatik 2“ und „Informatik 3“ ab, bei denen die Studierenden die ersten Web-Anwendungen programmieren und programmiert haben. Von der Anwendungsschicht ausgehend werden internet-typische Protokolle, Algorithmen, Verfahren und Hardware bis hin zur Leitungsebene vorgestellt und analysiert.

Im ersten Schritt werden das TCI/IP- und das ISO/OSI-Referenzmodell vorgestellt und typische Protokolle der Anwendungsebene, wie HTTP, FTP, SMTP, SNTP, und DNS, präsentiert [Kurose, Ross 2012]. Damit in Zusammenhang wird die Sniffing-Software WireShark vorgestellt [WireShark 2013] und die ersten HTTP- und FTP-Daten zwischen einem Server und einem Client abgehört. Typische Netzstrukturen wie 3- und 4-tier-Architekturen werden zusammen mit den Studenten ebenso entwickelt wie der Aufbau eines typischen Heim-Netzes mit DSL- und WLAN-Routern (LE1).

Im Anschluss daran werden die Protokolle TCP und UDP mit deren Unterschieden vorgestellt. Damit in Zusammenhang werden der 3-Wege-Handshake, der Stop-and-Wait- und der Sliding-Window-Algorithmus sowie die Überlastkontrolle von TCP erläutert [Tanenbaum 2003]. Als praktische Anwendungen dazu wird die Socket-Programmierung in Java vorgestellt und eingeübt [Calvert, Donahoo 2008] (LE2).

Auf der Netzwerkschicht werden IPv4, IPv6, ICMP, DHCP und NAT vorgestellt und mit WireShark analysiert. In diesem Zusammenhang werden Router als Hardware, die Bildung von Subnetzen sowie das Distance Vector Routing, das Link State Routing und der Spanning-Tree-Algorithmus als Routing-Verfahren beschrieben [Tanenbaum 2003] (LE3).

Auf der Bitübertragungsschicht und der Sicherungsschicht bzw. auf dem Ethernet-Layer werden Ethernet-Frames, Arten des Framings und grundlegende Begriffe und Verfahren der Fehlererkennung behandelt [Tanenbaum 2003]. Als Hardware werden in diesem Zusammenhang Switches und Hubs und deren Funktionsweise vorgestellt [Kurose, Ross 2012] (LE4).

Zur Übertragung der Daten zwischen verschiedenen Geräten werden die prinzipiellen Vermittlungsarten, Netztopologien und deren Eigenschaften besprochen. Zusätzlich werden Leistungsparameter für Netze definiert und anhand von typischen Beispielen berechnet [Tanenbaum 2003]. Zum Abschluss wird kurz auf die drahtlose Übertragung mittels WLAN, die damit verbundenen Standards und auf das Hidden-Station-Problem mit dessen Lösungsansätzen eingegangen [Tanenbaum 2003] (LE5).

Medienformen:

- Seminaristischer Unterricht mit PC-Beamer und Foliensatz sowie Demonstration der Analyse von Daten, die über ein Netzwerk versendet werden. Der Foliensatz steht zu Beginn der Veranstaltung zum Download bereit.

- Tafelanschrieb zur Unterstützung des Verständnisses für Kommunikation über ein Netzwerk.

Literatur:

- Calvert, Kenneth L.; Donahoo, Michael J. (2008): TCP/IP sockets in Java. Practical guide for programmers. 2. ed. Amsterdam, Heidelberg: Morgan Kaufmann.
- Kurose, James F.; Ross, Keith W. (2014): Computernetzwerke. Der Top-Down-Ansatz. 6. Aufl. München: Pearson.
- Tanenbaum, Andrew S. (2012): Computernetzwerke. 5. Aufl. München: Pearson.
- WireShark Network Sniffer: <<http://www.wireshark.org/>>

Modul:	Video	
Kürzel:	mkiB36	
Untertitel:		
Lehrveranstaltungen:	Vorlesung Praktikum	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Boris Terpinc	
Dozent(in):	Prof. Boris Terpinc	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 3. Semester	
Lehrform/SWS:	Vorlesung, Praktikum	4 (2+2) SWS
Arbeitsaufwand:	Präsenzstudium	60 Stunden
	Eigenstudium	90 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	mkiB15, mkiB16, mkiB26 müssen bestanden sein	
Empfohlene Voraussetzung :	keine	
Studien-/Prüfungsleistungen/ Prüfungsform:	Vorlesung: Hausarbeit, Praktikum: Projektarbeit	

Modulziele:

Das Modul Video baut auf den Veranstaltungen Grafik, Fotografie und Audio auf. Hier erlernte Kenntnisse werden im Videoseminar weitergeführt und vertieft. Wie werden Filme inhaltlich und technisch hergestellt? Eine Einführung ausgehend von der Geschichte der Filmtechnik bis zur modernen Video- und Filmaufnahmetechnik für Bild und Ton beantwortet die Eingangsfrage zunächst technisch. Die inhaltliche Arbeit von Autoren, Realisatoren und Regisseuren und die Teamarbeit in der Filmherstellung führen in die Praxis der Medien- und Kommunikationsbranche ein. Im Praktikum bilden Übungen im Filmschnitt und in Ton- und Bildaufnahme die Grundlagen. Darauf aufbauend folgt die Realisierung eines kurzen Videofilms nach inhaltlichen Vorgaben. Das Ziel ist, ein Video selbständig in Gruppenarbeit herzustellen, wobei im Team das Zusammenspiel spezifischer Rollen geübt wird: Drehbuch-Autor/Regie, Bildaufnahmetechnik, Tonaufnahmetechnik, Lichttechnik, Montage, Tonmischung, Aufnahmeleitung und Endbearbeitung (Titelgrafik, Farbkorrektur).

Angestrebte Lernergebnisse:

Kenntnisse:

- Studenten lernen die Grundlagen derameratechnik, Tontechnik, Lichttechnik und der digitalen Schnittprogramme.
- Sie kennen Audio/Video-Bildformate und deren Datentransfer.
- Sie entwickeln eigene Filmideen und setzen diese in szenischen Kurzfilmen um.
- Sie durchlaufen alle Produktionsschritte in der Filmherstellung.
- Sie kennen die Regeln der Teamarbeit, Arbeitsteilung und Teamleitung.
- Sie können Grundlagen audiovisueller Gestaltungsmittel nutzen.
- Sie kennen arbeits- und sicherheitstechnische Richtlinien und Maßnahmen.
- Sie kennen die Grundbegriffe journalistischen Schreibens.

Fertigkeiten:

Die Studenten recherchieren zu einem vorgegebenen Thema und entwickeln daraus eine Filmidee (Exposé). Weiterführende Recherchen dienen dazu, ein Drehbuch zu erarbeiten und später, je nach Genre, ein Storyboard anzufertigen. Parallel zum Drehbuch lernen sie technische Fertigkeiten: Filmschnitt, Kameraführung, Film-Tonaufnahme, Lichtgestaltung, Tonmischung, Titelbearbeitung und Fertigstellung für entsprechende Anwenderformate. Die Teamarbeit steht im Fokus der Ausbildung. Alle Arbeitsschritte der jeweiligen Teams werden von Dozenten und Assistenten im Laufe des Seminars begleitet. Dabei können die Studenten in einem Team verschiedene Rollen wählen, die sie hauptverantwortlich übernehmen: Idee und Regie, Kamera, Ton, Schnitt, Aufnahmeleitung. Jedes Team besteht aus mindestens 4 Mitgliedern. Als Hausarbeit ist ein Aspekt aus der Videoherstellung zu vertiefen, wobei Form und Zielgruppe eines journalistischen Genres vorgegeben werden.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage:

LE#	Lernergebnis (LE)	Geprüft durch
LE1	Eine Filmidee zu entwickeln, vom Exposé bis zum Storyboard.	Hausarbeit
LE2	Professionelle Filmaufnahmetechniken für Bild, Ton und Schnitt handwerklich auszuführen.	Hausarbeit
LE3	Gestalterische und journalistische Mittel zu differenzieren.	Hausarbeit
LE4	Verschiedene technische Geräte und Aufnahmetechniken zu unterscheiden.	Artefakt
LE5	Aufnahme und Verarbeitungsformate für Bild und Ton mit verschiedenen technischen Standards zu bewerten und anzuwenden.	Artefakt
LE6	Die erworbenen Kenntnisse eigenverantwortlich in einer Videoproduktion umzusetzen und sich als Team zu organisieren.	Artefakt
LE7	Einen journalistischen Artikel zum Thema der Videoherstellung oder zu einem spezifischen technischen oder inhaltlichen Aspekt zu verfassen.	Hausarbeit

Inhalt:

Die Studenten erlernen theoretische Grundlagen der Video- und Filmherstellung. Dabei werden in der Vorlesung drei grundlegende Ausrichtungen behandelt.

a) Gestaltung: Inhaltliche- und optische Recherche, das Verfassen von Exposé, Treatment, Drehbuch und Storyboard (LE1) als Grundlagen zur inhaltlichen Filmkonzeption. Bildgestaltung und die optische (szenische) Auflösung (LE2) dienen als Handwerkszeug der Filmerzählung, Filmdramaturgie und Filmkonzeption (LE3).

b) Technik: Die Geschichte der Film- und Kameratechnik führt zum Verstehen der Bild- und Tonaufnahmetechnik in der Film- und Videoindustrie (LE4). Der Videoschnitt erörtert die Frage, wie und warum Film funktioniert. Grundkenntnisse der Kamera, Beleuchtungstechnik, Tonaufnahmetechnik und Tonmischung (LE5) werden thematisiert. Aspekte für Titeldbearbeitung und Farbkorrektur runden den Prozess der Videoherstellung ab.

c) Verantwortung: Organisation, Planung und Reflexion. Die Studenten lernen im Umgang mit Menschen, vor und hinter der Kamera, ein Team zu führen, in einem Team zu arbeiten (LE6) und Darsteller vor der Kamera einzuweisen. Das Thema journalistisches Schreiben behandelt (LE7) Schreibtechniken und handwerkliche Grundlagen für das Verfassen von Artikeln, basierend auf dem Presse- und Urheberrecht. Das Persönlichkeitsrecht führt zur Diskussion journalistischer, rechtlicher und ethischer Fragen des Mediums Film.

Das Praktikum beginnt mit Übungsaufgaben zum Filmschnitt. Nach der Einarbeitung in die Videotechnik folgen Übungen mit Bild- und Tonaufnahmen. Grundlagen der Lichtgestaltung schließen die technische Einführung ab. Ziel ist, die Realisierung eines Kurzfilmes, wobei die Kreativität und Ideenreichtum gefördert werden. Die Studenten realisieren das Video eigenverantwortlich, das sie selbst inhaltlich und technisch konzipiert haben. Die Videos werden innerhalb eines Semesters fertig gestellt. Es ist Raum für individuelle Orientierung innerhalb der Praktika vorhanden: sei es sich in die Schnitttechnik zu vertiefen, sich auf die Kameraarbeit zu konzentrieren, für die Tonaufnahmen verantwortlich zu sein oder die Organisation und das Management zu übernehmen oder aber auch in die inhaltliche Ausarbeitung (Recherche und Drehbuch) zu gehen.

Medienformen:

Vorlesung mit audiovisuellen Beispielen und begleitendem Praktikum. Im Praktikum bearbeiten die Studierenden in Gruppen und als feste Teams ihre Übungsaufgaben und Videoprojekte. Dabei werden sie von Lehrbeauftragten und Assistenten, die aus der Filmbranche kommen, unterstützt. Alle notwendigen technischen Geräte, wie Videokameras, Tonaufnahmegeräte, Mikrofone, Filmlicht und Systemkameras stehen als Lehrmaterial zur Verfügung, eine technisch professionelle Grundlage für die Ausbildung. Die Videotechnik bedient sich der gängigen High Definition - Formate (HDTV) mit aktuellen Speichertechniken. Darüber hinaus gibt es Tonmischpulte für Außenaufnahmen und Mikrofone für Originalton Filmaufnahmen, sowie Filmlicht für Innen- und Außenaufnahmen. Die Medienlabore bestehen aus 10 Einzelschnitträumen mit digitalen Schnittprogrammen, einem TV-Studio für Objekt- und Studioaufnahmen in Blue-Box-Technik und mit professionellem Studioliicht (ARRI) sowie einem Tonstudio für Musik-, Sprachaufnahmen und für Film-Tonmischungen.

Literatur:

- Cioffi, Frank (2006): Kreatives Schreiben für Studenten & Professoren. Ein praktisches Manifest. Dt. Erstausg. Berlin: Autorenhaus-Verl.
- Dunker, Achim (2008): "Die chinesische Sonne scheint immer von unten". Licht- und Schattengestaltung im Film. 5., überarb. Aufl. Konstanz: UVK-Verl.-Ges (Praxis Film, Bd. 47).

- Eick, Dennis (2005): Exposee, Treatment und Konzept. Konstanz: UVK (Praxis Film, 26).
- Hoffmann-Walbeck, Thomas; Zimmermann, Gottfried; Hedler, Marko; Homann, Jan-Peter; Henka, Alexander; Riegel, Sebastian et al. (2013): Standards in der Medienproduktion. Berlin, Heidelberg: Springer Vieweg (X.media.press).
- Kandorfer, Pierre (2010): Das Lehrbuch der Filmgestaltung. Theoretisch-technische Grundlagen der Filmkunde. 7., überarb. Aufl. Berlin: Schiele & Schön.
- Katz, Steven D. (2010): Die richtige Einstellung. Shot by shot; zur Bildsprache des Films; das Handbuch. Dt. Erstausg., 6. Aufl. Frankfurt am Main: Zweitausendeins.
- Reil, Andreas A. (2001): Lexikon Film, TV, Fernsehen, Video & Internet. [3000 Begriffe verständlich erklärt]. 4. Aufl. Wesseling: Reil (Media-Handbücher).
- Schneider, Wolf (2013): Deutsch! Das Handbuch für attraktive Texte. 5. Aufl. Reinbek bei Hamburg: Rowohlt-Taschenbuch-Verl (rororo, 61993 : Sachbuch).
- Vogel, Andreas (c2012): Handbuch HD-Produktion 2013. Transfer Media. [überarb. und aktualisiert]. Berlin: Schiele & Schön.

Modul:	Datenbanksysteme	
Kürzel:	mkiB41	
Untertitel:		
Lehrveranstaltungen:	Vorlesung	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Dr. Peter Hertkorn	
Dozent(in):	Prof. Dr. Peter Hertkorn	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 4. Semester	
Lehrform/SWS:	Vorlesung	4 SWS
Arbeitsaufwand:	Präsenzstudium	60 Stunden
	Eigenstudium	90 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung:	mkiB13, mkiB14, mkiB23, mkiB24, mkiB31, mkiB32	
Studien-/Prüfungsleistungen/ Prüfungsform:	Vorlesung: Klausur	

Modulziele:

Die Studierenden erlangen Kenntnisse über die Funktionsweise von Datenbanksystemen und unterschiedliche Datenbanktechnologien. Sie verstehen die zugrunde liegenden Prinzipien, Methoden und Techniken und können die theoretischen Kenntnisse in der Praxis anwenden. Im weiteren Verlauf des Studiums soll mit dem erfolgreichen Bestehen des Moduls gewährleistet werden, dass die Studierenden einen systematischen Datenbankentwurf durchführen, Modellierungsentscheidungen abwägen und Datenbanken mit Hilfe von Datenbank- und Programmiersprachen aufbauen und nutzen können.

Angestrebte Lernergebnisse:

Kenntnisse:

- Aufgaben und Ziele eines Datenbanksystems erläutern können.
- Die Architektur von Datenbanksystemen darstellen können.
- Die Phasen des Datenbankentwurfs beschreiben können.

- Verschiedene Datenmodelle kennen und die Unterschiede zwischen den Datenmodellen erklären können.
- Methoden zur Modellierung von Zusammenhängen der realen Welt kennen.
- Methoden zur Abbildung des semantischen Datenmodells auf ein relationales Modell kennen.
- Ursachen für Datenanomalien erklären können und Verfahren zu deren Vermeidung kennen.
- Konzepte und Elemente von Datenbanksprachen kennen.
- Basiseigenschaften von Transaktionen erläutern können.
- Probleme durch Nebenläufigkeit darstellen können und Verfahren zu deren Behandlung kennen.
- Methoden zum Zugriff auf eine Datenbank aus einem Anwendungsprogramm kennen.
- Eigenschaften objektrelationaler Datenbanken erklären können.
- Methoden zur Speicherung semistrukturierter Daten kennen.
- Konzepte von neueren Entwicklungen wie NoSQL-Datenbanken beschreiben und die Unterschiede zu relationalen Datenbanksystemen erläutern können.

Fertigkeiten:

Die Studierenden analysieren die Anforderungen für eine gegebene Problemstellung und erstellen daraus ein semantisches Datenmodell. Aus dem semantischen Datenmodell leiten sie ein relationales Modell ab. Die Studierenden können die Qualität des relationalen Modells beurteilen und wenden Verfahren zur Vermeidung von Datenanomalien an. Sie erstellen und modifizieren relationale Datenbankschemas mittels Datenbanksprachen und formulieren Anfragen sowie Änderungen an die Datenbank. Die Studierenden wenden unterschiedliche Verfahren zur Steuerung nebenläufiger Transaktionen an. Die Studierenden erstellen Datenbankschemas und Anfragen für objektrelationale Datenbanken sowie Anfragen für semistrukturierte Daten anhand der XML-Erweiterungen des relationalen Modells.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage:

LE#	Lernergebnis (LE)	Geprüft durch
LE1	Unterschiedliche Methoden für den systematischen Entwurf von Datenbanken anzuwenden.	Klausur
LE2	Modellierungsalternativen bei der Erstellung der Datenbanken zu bewerten.	Klausur
LE3	Datenbanken für unterschiedliche Datenmodelle mit Datenbanksprachen zu erstellen.	Klausur
LE4	Für gegebene Anforderungen Anfragen an die Datenbank zu formulieren.	Klausur
LE5	Alternative Möglichkeiten bei Anfragen an die Datenbank zu bewerten und hinsichtlich Performanz zu beurteilen.	Klausur
LE6	Transaktionen im Hinblick auf Probleme durch Nebenläufigkeit zu analysieren und geeignete Verfahren zur Mehrbenutzersynchronisation anzuwenden.	Klausur
LE7	Aktuelle Entwicklungen im Bereich Datenbanksysteme zu beurteilen und sich anzueignen.	Klausur

Inhalt:

In der Vorlesung werden die Studierenden über den Einsatz von Datenbanksystemen und das grundlegende Architekturmodell an das Themengebiet herangeführt. Entsprechend der Phasen des Datenbankentwurfs werden die einzelnen Entwurfsschritte systematisch anhand eines Fallbeispiels durchgeführt (LE1). Zur semantischen Datenmodellierung wird das Entity-Relationship-Modell verwendet (LE1, LE2). Für das relationale Modell werden sowohl die Theorie als auch praktische Entwurfsregeln behandelt (LE1, LE2). Mit der Datenmanipulations- und Definitionssprache SQL werden Datenbankschemas erstellt und Anfragen an die Datenbank entwickelt (LE3-5). Zum Verständnis der Datenbankfunktionalität werden Transaktionskonzepte sowie Synchronisationsmechanismen untersucht (LE6). Neben klassischen relationalen Datenbanken werden objektrelationale Datenbanken und die XML-Erweiterungen des relationalen Modells behandelt (LE3, LE4). Des Weiteren werden neuere Entwicklungen wie NoSQL-Datenbanken vorgestellt und deren Eigenschaften mit denen der relationalen Datenbanksysteme verglichen (LE7). Der Zugriff auf Datenbanken aus einer Anwendung heraus wird vorgestellt und an Beispielprogrammen erläutert.

Medienformen:

Das Lehrmaterial besteht aus einem Folienskript, das in elektronischer Form vorliegt, Übungsblättern sowie Programmbeispielen. Seminaristischer Unterricht mit Tafelanschrieb, PC-Beamer und Präsentationsfolien, bei dem Beispiele zu den theoretischen Inhalten veranschaulicht werden sowie Demonstration von Beispielprogrammen und interaktiver Programmentwicklung.

Literatur:

- Beaulieu, Alan (2009): Learning SQL. 2nd ed (Online-Ausg.). Sebastopol: O'Reilly Media (EBL-Schweitzer).
- Connolly, Thomas (2015): Database Systems: A Practical Approach to Design. 6. Aufl. Harlow: Pearson Education Limited.
- Date, Chris J. (2004): An introduction to database systems. 8. ed., internat. ed. Boston, München: Pearson Addison Wesley.
- Elmasri, Ramez; Navathe, Sham (2010): Fundamentals of database systems. 6th ed. Upper Saddle River, N.J., Harlow: Pearson Education.
- Garcia-Molina, Hector; Ullman, Jeffrey D.; Widom, Jennifer (2009): Database systems. The complete book. 2. ed., internat. ed. Upper Saddle River, NJ: Pearson Prentice Hall.
- Kemper, Alfons; Eickler, André (2013): Datenbanksysteme. Eine Einführung. 9., erw. und aktual. Aufl. München: Oldenbourg.
- Kemper, Alfons; Wimmer, Martin (2012): Übungsbuch Datenbanksysteme. 3., aktualisierte und erw. Aufl. München: Oldenbourg.
- Saake, Gunter; Sattler, Kai-Uwe; Heuer, Andreas (2013): Datenbanken. Konzepte und Sprachen. 5. Aufl. Heidelberg, München, Landsberg, Frechen, Hamburg: mitp.
- Sadalage, Pramod J.; Fowler, Martin (2012, c2013): NoSQL distilled. A brief guide to the emerging world of polyglot persistence. Upper Saddle River, NJ: Addison-Wesley. Online verfügbar unter <http://proquest.tech.safaribooksonline.de/9780133036138>.
- Vonhoegen, Helmut (2013): Einstieg in XML. Grundlagen Praxis Referenz; [für Anwendungsentwicklung und E-Publishing; Transformation Formatierung; Schnittstellen; XML Schema DTD XSLT CSS XSL XPath DOM SAX SOAP XQuery; XForms HTML5 EPUB]. 7., aktualisierte und erw. Aufl. Bonn: Galileo Press (Galileo Computing).

- Vossen, Gottfried (2008): Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme. 5., überarb. und erw. Aufl. München, Wien: Oldenbourg.

Modul:	Datenbanksysteme Praktikum	
Kürzel:	mkiB42	
Untertitel:		
Lehrveranstaltungen:	Praktikum	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Dr. Peter Hertkorn	
Dozent(in):	Prof. Dr. Peter Hertkorn	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 4. Semester	
Lehrform/SWS:	Vorlesung	2 SWS
Arbeitsaufwand:	Präsenzstudium	30 Stunden
	Eigenstudium	120 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung	mkiB13, mkiB14, mkiB23, mkiB24, mkiB31, mkiB32	
Studien-/Prüfungsleistungen/ Prüfungsform:	Praktikum: Praktikum	

Modulziele:

Die Studierenden erlangen Kenntnisse über die Funktionsweise von Datenbanksystemen und unterschiedliche Datenbanktechnologien. Sie verstehen die zugrunde liegenden Prinzipien, Methoden und Techniken und können die theoretischen Kenntnisse in der Praxis anwenden. Im weiteren Verlauf des Studiums soll mit dem erfolgreichen Bestehen des Moduls gewährleistet werden, dass die Studierenden einen systematischen Datenbankentwurf durchführen, Modellierungsentscheidungen abwägen und Datenbanken mit Hilfe von Datenbank- und Programmiersprachen aufbauen und nutzen können.

Angestrebte Lernergebnisse:

Kenntnisse:

- Arten von Anforderungen an ein Datenbanksystem erläutern können.
- Die Phasen des Datenbankentwurfs beschreiben können.
- Verschiedene Datenmodelle kennen und die Unterschiede zwischen den Datenmodellen erklären können.

- Methoden zur Modellierung von Zusammenhängen der realen Welt kennen.
- Methoden zur Abbildung des semantischen Datenmodells auf ein relationales Modell kennen.
- Ursachen für Datenanomalien erklären können und Verfahren zu deren Vermeidung kennen.
- Konzepte und Elemente von Datenbanksprachen kennen.
- Probleme durch Nebenläufigkeit darstellen können und Verfahren zu deren Behandlung kennen.
- Methoden zum Zugriff auf eine Datenbank aus einem Anwendungsprogramm kennen.
- Erweiterungen des objekt-relationalen Modells kennen.
- Methoden zur Abbildung semistrukturierter Daten erklären können.
- Konzepte von neueren Entwicklungen wie NoSQL-Datenbanken beschreiben und die Unterschiede zu relationalen Datenbanksystemen erläutern können.

Fertigkeiten:

Die Studierenden formulieren Anforderungen an ein Datenbanksystem und erstellen daraus ein semantisches Datenmodell. Aus dem semantischen Datenmodell leiten sie ein relationales Modell ab und wenden Verfahren zur Normalisierung an. Sie erstellen und modifizieren relationale Datenbankschemas mittels Datenbanksprachen und formulieren Anfragen sowie Änderungen an die Datenbank. Sie analysieren die Anfragen hinsichtlich ihrer Performanz und wenden unterschiedliche Methoden zur Optimierung an. Die Studierenden wenden verschiedene Verfahren zur Steuerung nebenläufiger Transaktionen an. Sie entwickeln Lösungen, um den Zugriff auf die Datenbank aus einem Anwendungsprogramm realisieren zu können. Die Studierenden erstellen Anfragen für semistrukturierte Daten anhand der XML-Erweiterungen des relationalen Modells. Sie formulieren Anfragen an NoSQL-Datenbanken und greifen aus Anwendungsprogrammen auf diese zu.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage:

LE#	Lernergebnis (LE)	Geprüft durch
LE1	Eine Datenbank über die verschiedenen Phasen des Datenbankentwurfs systematisch zu entwerfen.	Artefakt
LE2	Datenbanken mit Hilfe von SQL zu erstellen und Anfragen an die Datenbank zu formulieren.	Artefakt
LE3	SQL-Anfragen zu analysieren und im Hinblick auf Performanz zu optimieren.	Artefakt
LE4	Transaktionen im Hinblick auf Probleme durch Nebenläufigkeit zu beurteilen und Verfahren zur Mehrbenutzersynchronisation anzuwenden.	Artefakt
LE5	Den Zugriff auf Datenbanken aus einem Anwendungsprogramm realisieren zu können.	Artefakt
LE6	Professionelle Werkzeuge für den Entwurf, den Aufbau und die Nutzung von Datenbanken einsetzen zu können.	Artefakt
LE7	Probleme und Grenzen, die bei der Datenbankentwicklung entstehen, einzuschätzen.	Artefakt

Inhalt:

Im Praktikum wenden die Studierenden die in der Vorlesung erworbenen Kenntnisse selbständig beim Lösen von Übungsaufgaben sowie bei der Erstellung und Nutzung von Datenbanken an. Entsprechend der Phasen des Datenbankentwurfs wird eine Datenbank ausgehend von einer Aufgabenbeschreibung mit Hilfe geeigneter Werkzeuge in kleinen Übungseinheiten entworfen (LE1, LE6). Bei der Bearbeitung von Problemstellungen werden dabei verschiedene Alternativen betrachtet und miteinander verglichen (LE7). Die Erstellung des Datenbankschemas und die Entwicklung von Anfragen an die Datenbank erfolgt interaktiv und Skript-basiert mit geeigneten Werkzeugen auf Basis eines Standard-Datenbanksystems (LE2, LE6). Dabei werden unterschiedliche Arten der Optimierung von Anfragen analysiert und im Hinblick auf die Laufzeit bewertet (LE3). Zur Verdeutlichung der Probleme nebenläufiger Anfragen werden konkurrierende Transaktionen im Multiuserbetrieb bei verschiedenen Isolationsstufen untersucht (LE4). Die Unterschiede bei der Speicherung von semistrukturierten und multistrukturierten Daten werden anhand der XML-Erweiterungen des relationalen Modells und NoSQL-Datenbanken verdeutlicht. Für den Zugriff auf Datenbanken aus einem Anwendungsprogramm entwickeln die Studierenden Lösungen unter Verwendung von Entwicklungsumgebungen (LE5, LE6).

Medienformen:

Die Studierenden bearbeiten individuell oder in Gruppen Übungsaufgaben zum Themengebiet Datenbanksysteme. Die Studierenden entwickeln mit Hilfe von Werkzeugen Modelle und Programme. Betreuung durch den Dozenten. Eine umfangreichere Semesterarbeit ist über mehrere Wochen hinweg zu bearbeiten, um die Studierenden an größere Aufgaben heranzuführen.

Literatur:

- Beaulieu, Alan (2009): Learning SQL. 2nd ed (Online-Ausg.). Sebastopol: O'Reilly Media (EBL-Schweitzer).
- Connolly, Thomas (2015): Database Systems: A Practical Approach to Design. 6. Aufl. Harlow: Pearson Education Limited.
- Date, Chris J. (2004): An introduction to database systems. 8. ed., internat. ed. Boston, München: Pearson Addison Wesley.
- Elmasri, Ramez; Navathe, Sham (2010): Fundamentals of database systems. 6th ed. Upper Saddle River, N.J., Harlow: Pearson Education.
- Garcia-Molina, Hector; Ullman, Jeffrey D.; Widom, Jennifer (2009): Database systems. The complete book. 2. ed., internat. ed. Upper Saddle River, NJ: Pearson Prentice Hall.
- Kemper, Alfons; Eickler, André (2013): Datenbanksysteme. Eine Einführung. 9., erw. und aktual. Aufl. München: Oldenbourg.
- Kemper, Alfons; Wimmer, Martin (2012): Übungsbuch Datenbanksysteme. 3., aktualisierte und erw. Aufl. München: Oldenbourg.
- Saake, Gunter; Sattler, Kai-Uwe; Heuer, Andreas (2013): Datenbanken. Konzepte und Sprachen. 5. Aufl. Heidelberg, München, Landsberg, Frechen, Hamburg: mitp.
- Sadalage, Pramod J.; Fowler, Martin (2012, c2013): NoSQL distilled. A brief guide to the emerging world of polyglot persistence. Upper Saddle River, NJ: Addison-Wesley. Online verfügbar unter <http://proquest.tech.safaribooksonline.de/9780133036138>.
- Vonhoegen, Helmut (2013): Einstieg in XML. Grundlagen Praxis Referenz; [für Anwendungsentwicklung und E-Publishing; Transformation Formatierung;

Schnittstellen; XML Schema DTD XSLT CSS XSL XPath DOM SAX SOAP XQuery; XForms HTML5 EPUB]. 7., aktualisierte und erw. Aufl. Bonn: Galileo Press (Galileo Computing).

- Vossen, Gottfried (2008): Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme. 5., überarb. und erw. Aufl. München, Wien: Oldenbourg.

Modul:	Softwaretechnik 2	
Kürzel:	mkiB43	
Untertitel:		
Lehrveranstaltungen:	Vorlesung Praktikum	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Dr. Peter Hertkorn	
Dozent(in):	Prof. Dr. Peter Hertkorn	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 4. Semester	
Lehrform/SWS:	Vorlesung, Praktikum	4 (2+2) SWS
Arbeitsaufwand:	Präsenzstudium	60 Stunden
	Eigenstudium	90 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung	mkiB31, mkiB32, mkiB33	
Studien-/Prüfungsleistungen/ Prüfungsform:	Vorlesung: Klausur Praktikum: Praktikum	

Modulziele:

Die Studierenden erlangen Kenntnisse über die Analyse und den Entwurf von Software-Systemen. Sie verstehen die zugrunde liegenden Prinzipien, Methoden und Techniken und können die theoretischen Kenntnisse in der Praxis anwenden. Als Erweiterung zum Modul Softwaretechnik 1 werden die Kenntnisse in der Modellierung vertieft und beim Entwurf von Software-Systemen angewendet. Im weiteren Verlauf des Studiums soll mit dem erfolgreichen Bestehen des Moduls gewährleistet werden, dass die Studierenden die Bausteine von Softwarearchitekturen kennen, Architekturprinzipien verstehen, Architektur- und Entwurfsmuster anwenden sowie Methoden der modellgetriebenen Softwareentwicklung einsetzen können.

Angestrebte Lernergebnisse:

Kenntnisse:

- Konzepte und Methoden der Modellierung von Software-Systemen kennen.

- Analysemuster beschreiben können.
- Unterschiedliche Sichten auf Softwarearchitekturen darstellen können.
- Bausteine von Softwarearchitekturen kennen.
- Architekturprinzipien erläutern können.
- Architekturmuster beschreiben können.
- Einflussfaktoren auf die Softwarearchitektur darstellen können.
- Entwurfsmuster beschreiben können.
- Methoden der modellgetriebenen Softwareentwicklung kennen.
- Elemente von domänenspezifischen Sprachen kennen.

Fertigkeiten:

Die Studierenden analysieren Problemstellungen und wenden die erlernten Methoden für die Modellierung und den Entwurf an. Sie verwenden bewährte Konzepte und Ansätze in Form von Analyse-, Architektur- und Entwurfsmustern zur Entwicklung eigener Lösungen. Sie vergleichen unterschiedliche Lösungen und stellen deren Vor- und Nachteile dar. Die Studierenden wenden Methoden der modellgetriebenen Softwareentwicklung an und erstellen domänenspezifische Sprachen mit Hilfe geeigneter Werkzeuge.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage:

LE#	Lernergebnis (LE)	Geprüft durch
LE1	Unterschiedliche Methoden für die Modellierung und den Entwurf von Software-Systemen anzuwenden.	Klausur, Artefakt
LE2	Alternativen bei der Modellierung und dem Entwurf von Software-Systemen zu bewerten.	Klausur, Artefakt
LE3	Unterschiedliche Softwarearchitekturen zu analysieren und zu beurteilen.	Klausur, Artefakt
LE4	Verfahren der modellgetriebenen Softwareentwicklung zu analysieren und deren Vor- und Nachteile bewerten zu können.	Klausur, Artefakt
LE5	Professionelle Werkzeuge für die Modellierung und den Entwurf von Software-Systemen einsetzen zu können.	Artefakt
LE6	Probleme und Grenzen, die bei der Modellierung und dem Entwurf von Software-Systemen entstehen, einzuschätzen.	Artefakt
LE7	Aktuelle Entwicklungen im Bereich Softwaretechnik beurteilen und sich aneignen können.	Artefakt

Inhalt:

In der Vorlesung werden die in Softwaretechnik 1 erhaltenen Kenntnisse zur Modellierung von Software-Systemen vertieft und der Einsatz von Modellierungssprachen im Hinblick auf die modellgetriebene Softwareentwicklung erweitert (LE1). Verschiedene Alternativen der Modellierung werden diskutiert und bewährte Ansätze wie Analysemuster besprochen (LE2). Beim Entwurf von Software-Systemen werden verschiedene Sichten in der Softwarearchitektur, Prinzipien des Architekturentwurfs sowie Bausteine von Architekturen und übergreifende Themen behandelt (LE3). Die Gliederung in überschaubare Einheiten wird am Beispiel von Komponentenmodellen dargestellt. Bewährte Lösungen in Form von Architektur- und Entwurfsmustern werden detailliert besprochen (LE1, LE2). Die

Studierenden lernen darüber hinaus Techniken kennen, mit denen automatisiert aus formalen Modellen lauffähige Software erzeugt werden kann (LE4).

Im Praktikum wenden die Studierenden die in der Vorlesung erworbenen Kenntnisse selbständig beim Lösen von Übungsaufgaben sowie bei der Erstellung von Programmen an. Die Erstellung von Modellen und Programmen erfolgt mit geeigneten Werkzeugen (LE5). Bei der Bearbeitung von Problemstellungen werden dabei verschiedene Alternativen betrachtet und miteinander verglichen (LE6). Die Studierenden recherchieren wissenschaftliche Literatur zum Thema und vertiefen die Vorlesungsinhalte eigenverantwortlich (LE7).

Medienformen:

Das Lehrmaterial besteht aus einem Folienskript, das in elektronischer Form vorliegt, Übungsblättern sowie Programmbeispielen. Das Modul umfasst eine Vorlesung mit begleitendem Praktikum. Seminaristischer Unterricht mit Tafelanschrieb, PC-Beamer und Präsentationsfolien, bei dem Beispiele zu den theoretischen Inhalten veranschaulicht werden, sowie Demonstration von Beispielprogrammen und interaktiver Programmentwicklung. Die Studierenden bearbeiten individuell oder in Gruppen Übungsaufgaben zum Themengebiet Softwaretechnik und entwickeln mit Hilfe von Werkzeugen Modelle und Programme. Betreuung durch den Dozenten.

Literatur:

- Balzert, Helmut (2011): Lehrbuch der Softwaretechnik Entwurf, Implementierung, Installation und Betrieb. 3. Aufl. Heidelberg: Spektrum Akademischer Verlag (SpringerLink : Bücher).
- Brügge, Bernd; Dutoit, Allen Henry (2014): Object-oriented software engineering. Using UML patterns and Java. 3. ed., international ed. Harlow, Essex: Pearson.
- Fowler, Martin (2010 [erschienen] 2011): Domain-specific languages. 1. print. Upper Saddle River, NJ, Munich: Addison-Wesley (A Martin Fowler signature book).
- Goll, Joachim; Dausmann, Manfred (2013): Architektur- und Entwurfsmuster der Softwaretechnik. Mit lauffähigen Beispielen in Java. Wiesbaden: Springer Vieweg (SpringerLink : Bücher).
- Larman, Craig (2005): Applying UML and patterns. An introduction to object-oriented analysis and design and iterative development. 3. ed. Upper Saddle River, NJ, [München]: Prentice Hall; Markt + Technik.
- Metsker, Steven John; Wake, William C. (2006): Design patterns in Java. Upper Saddle River, NJ, Munich: Addison-Wesley (The software patterns series).
- Oestereich, Bernd; Scheithauer, Axel (2013): Analyse und Design mit der UML 2.5. Objektorientierte Softwareentwicklung; [inkl. Poster mit UML-Notationsübersicht]. 11., umfassend überarb. und aktualisierte Aufl. München: Oldenbourg.
- Stahl, Thomas (2007): Modellgetriebene Softwareentwicklung. Techniken Engineering Management. 2., aktualisierte und erw. Aufl. Heidelberg: dpunkt-Verl.
- Starke, Gernot (2014): Effektive Softwarearchitekturen. Ein praktischer Leitfaden. 6., überarb. Aufl. München: Hanser.
- Vogel, Oliver; Arnold, Ingo; Chughtai, Arif; Ihler, Edmund; Kehrer, Timo; Mehlig, Uwe; Zdun, Uwe (2009): Software-Architektur. Grundlagen — Konzepte — Praxis. 2. Auflage. Heidelberg: Spektrum Akademischer Verlag (SpringerLink : Bücher).
- Völter, Markus (2013): DSL engineering. Designing implementing and using domain-specific languages. [S.I.]: CreateSpace Independent Publishing Platform.

- Winter, Mario (2005): Methodische objektorientierte Softwareentwicklung. Eine Integration klassischer und moderner Entwicklungskonzepte. 1. Aufl. Heidelberg: dpunkt-Verl.

Modul:	Verteilte Systeme	
Kürzel:	mkiB44	
Untertitel:		
Lehrveranstaltungen:	Vorlesung Praktikum	
Studiensemester:	jedes Semester	
Modulverantwortliche:	Prof. Dr. Natividad Martinez	
Dozent(in):	Prof. Dr. Natividad Martinez	
Sprache:	Deutsch, Englisch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 4. Semester	
Lehrform/SWS:	Vorlesung, Praktikum	4 (2+2) SWS
Arbeitsaufwand:	Präsenzstudium Eigenstudium	60 Stunden 90 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	Keine	
Empfohlene Voraussetzung	mkiB31, mkiB32, mkiB33, mkiB34, mkiB35	
Studien-/Prüfungsleistungen/ Prüfungsform:	Vorlesung und Praktikum: Continuous Assessment	

Modulziele:

Die Studierenden erhalten Kenntnisse über Systeme und Architekturen zur Nutzung verteilter Rechnerressourcen. Sie sind in der Lage verteilte Anwendungen zu programmieren und besitzen Kenntnisse grundlegender verteilter Algorithmen. Des Weiteren kennen Sie die Vor- und Nachteile von Technologien zur Erstellung verteilter Anwendungen und können diese erklären. Sie besitzen die Kompetenz zur Auswahl einer geeigneten verteilten Technologie für ein gegebenes Problem. Insbesondere wird ein Schwerpunkt auf die Webtechnologien gesetzt. Die Studierenden lernen den Entwurf und die Programmierung von Webanwendungen und wenden die in einem Projekt prototypisch an.

Das Modul bindet zusammen viele Fertigkeiten und Kompetenzen, die die Studierenden bereits in anderen Vorlesungen erworben haben: Software entwickeln und programmieren, systemische Aspekte aus Betriebssysteme, Datenbanken und Internetworking, zu einem gesamten, end-to-end System. Das Modul ist die Basis für die weiterführende Vorlesungen Mobile Computing und Cloud Computing.

Angestrebte Lernergebnisse:

Kenntnisse:

Die Studierenden kennen die Eigenschaften und Struktur eines verteilten Systems. Sie kennen die typische Architektur eines Middlewares, so wie das Paradigma der Fernmethodenaufrufe und deren Implementierung.

Die Studierenden stellen die Eigenschaften der Komponenten-basierten verteilten Architekturen am Beispiel von unternehmensbasierten verteilten Architekturen.

Die Studierenden kennen das Prinzip der Service-Oriented Architektur (SOA) am Beispiel der Web Services und klassifizieren diese unter REST Architekturen und SOAP-basierten Web Services.

Weiterhin kennen die Studierenden die wichtigsten Technologien um die Client- und Server-Seite einer Webanwendung zu entwerfen.

Fertigkeiten:

Die Studierenden modellieren und setzen ein verteiltes System auf dem Java-RMI-Middleware in Betrieb. Dabei benutzen sie die basische Kommunikationsmechanismen und -muster.

Die Studierenden teilen eine Unternehmenswebanwendung in Prozesse und Entitäten auf, um die gegebenen Anforderungen zu erfüllen. Sie entwickeln Unternehmenswebanwendungen in einer mehrschichtigen Architektur und binden in dieser einfache Zugriffe auf Datenbanken.

Die Studenten entwerfen Web Services und beurteilen, welche Art (REST oder SOAP) sich am besten eignet.

Die Studierenden wenden die herkömmlichen Technologien zu Informationsdarstellung und Webprogrammierung an. Dabei erkennen sie Aspekte der Sicherheit, Datenschutz, Effizienz und Benutzbarkeit.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage:

LE#	Lernergebnis (LE)	Geprüft durch
LE1	Unterschiedliche verteilte Architekturen zu analysieren und zu beurteilen.	Klausur
LE2	Lösungsvorschläge für Beispielszenarien auf der Basis der gelernten verteilten Technologien zu erarbeiten.	Übungen
LE3	Ein vollständiges web-basiertes verteiltes System unter bestimmten Anforderungen und Randbedingungen zu entwerfen.	Projektarbeit
LE4	In Team zu arbeiten um komplexe Aufgaben zu lösen.	Projektarbeit
LE5	Moderne Entwicklungsumgebungen und Werkzeuge zu benutzen.	Projektarbeit
LE6	Selbständig Information über aktuelle Entwicklungen der Disziplin Verteilte Systeme zu suchen, zu beurteilen und wissenschaftlich zu kommunizieren.	Ausarbeitung

LE7	Themen der Disziplin Verteilte Systeme unter Verwendung der Fachsprache kompetent zu präsentieren und zu diskutieren.	Referat
-----	---	---------

Inhalt:

Die Vorlesung Verteilte Systeme integriert viele der Kompetenzen, die die Studierenden bereits erworben haben in den Gebieten Programmierung, Softwareentwicklung und Datenbanken. Dabei wird der Inhalt sehr umfangreich und anwendungsorientiert. Deswegen wurde die Prüfungsform „Continuous Assessment“ ausgewählt, um die Studierenden mit unterschiedlichen Artefakten begleiten zu können. Die Vorlesung teilt sich in vier Gebiete auf. Zu jedem Gebiet gehört eine theoretische Einführung, deren Verständnis mit kurzen Testaten geprüft wird (LE1) zusammen mit Übungen (LE2), die die Studierenden allein oder in Gruppen lösen. Anschließend gibt es praktische Laboraufgaben (LE5), die in Gruppen gelöst werden und zu einer Projektarbeit gebündelt werden (LE3, LE4). In der zweiten Hälfte der Vorlesung bereiten die Studierenden eine Ausarbeitung über ein aktuelles Thema der Disziplin vor (LE6) und tragen Ihre Erkenntnisse vor (LE7). Die Gebiete sind:

1. Grundsätze und Architekturen von verteilten Systemen, Middleware für verteilte Systeme und Remote Method Invocation (RMI) [Coulouris].
2. Web Programmierung: Client-side versus server-side Programmierung [Sebesta].
3. Service-Oriented Architectures und Web Services: Eigenschaften und Vergleich von REST und SOAP Architekturen [Kalin].
4. Verteilte komponenten-basierten Plattformen für Unternehmensanwendungen [Wetherbee].

Medienformen:

Abhängig von Inhalt und Kompetenz werden unterschiedliche Medienformen ausgewählt. Einige Themen werden klassisch mit Folienskripten behandelt, die mit dem Beamer projiziert werden und durch den Einsatz der Tafel vertieft, erklärt und veranschaulicht werden können. In einem seminaristischen Unterrichtstil werden die Studierende einzeln oder in Gruppen Themen mit Hilfe von ausgesuchten Literaturreferenzen bearbeiten und vortragen. Das Modul umfasst eine Vorlesung mit begleitendem Praktikum. Dafür werden die Studierenden Hinweise über die notwendige Installation und die Anforderungen an die Systeme erhalten, die sie prototypisch unter Betreuung der Dozenten im Labor entwickeln sollten.

Literatur:

- Coulouris, George F. (2012): Distributed systems. Concepts and design. 5. ed., internat. ed. Boston, Munich: Addison-Wesley.
- Kalin, Martin (2013): Java Web services. Up and running. 2nd ed. Beijing: O'Reilly. Online verfügbar unter <http://proquest.tech.safaribooksonline.de/9781449373856>.
- Sebesta, Robert W. (2013): Programming the World Wide Web. 7. ed. Boston: Pearson.
- Wetherbee, Jonathan (2013): Beginning EJB 3. Java EE 7 edition. [Berkeley, CA]: Apress. Online verfügbar unter <http://proquest.tech.safaribooksonline.de/9781430246923>.
- Darüber hinaus aktuelle Artikel aus Fachjournalen und Konferenzen sowie Internet Ressourcen.

Modul:	IT-Sicherheit	
Kürzel:	mkiB45	
Untertitel:		
Lehrveranstaltungen:	Vorlesung Praktikum	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Dr. Michael Tangemann	
Dozent(in):	Prof. Dr. Michael Tangemann	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 4. Semester	
Lehrform/SWS:	Vorlesung, Praktikum	4 (2+2) SWS
Arbeitsaufwand:	Präsenzstudium	60 Stunden
	Eigenstudium	90 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung:	mkiB34, mkiB35	
Studien-/Prüfungsleistungen/ Prüfungsform:	Vorlesung: Klausur Praktikum: Praktikum	

Modulziel:

IT-Sicherheit ist ein typisches Querschnittsthema der Informatik. In den Veranstaltungen Betriebssysteme, Verteilte Systeme, Datenbanken und Internetworking werden bereits einzelne Aspekte im jeweiligen Kontext adressiert. Diese Veranstaltung gibt eine Gesamtschau auf das Gebiet, so dass bis dahin isolierte Aspekte im Zusammenhang verstanden und eingeordnet werden können.

Angestrebte Lernergebnisse:

Kenntnisse:

Die Teilnehmer werden für Schwachstellen informationstechnischer Systeme sowie deren Entstehung bei der Entwicklung und beim Betrieb sensibilisiert. Sie lernen die Ursachen der Verwundbarkeiten kennen, sowohl in Bezug auf den Einsatz unsicherer Programmier- und Makrosprachen sowie veralteter, unsicherer Betriebssysteme als auch in Bezug zum Betrieb der IT-Infrastruktur. Die Ansätze von Angreifern und Schadsoftware werden verstanden. Die

Studierenden sollen exemplarisch methodische Kenntnisse zur Analyse von Schadsoftware und Angreifertechniken sowie zur Erkennung von Verwundbarkeiten und deren Ausnutzung erwerben und anwenden können. Sie erwerben die Fähigkeit zur Anwendung von Sicherheitsprinzipien bei der Konfiguration von Sicherheitsmechanismen und bei der Implementierung von Anwendungen.

Fertigkeiten

Die Studenten sollen für die Bedrohungen von IT-Anlagen und Kommunikationsnetzen sensibilisiert werden. Sie sind nach Absolvieren des Kurses in der Lage, Angriffe zu identifizieren und zu klassifizieren, und sie können die aktuelle Gefährdungslage von IT-Anlagen einzuschätzen. Basierend auf der Kenntnis der Auswirkungen verschiedener Angriffe können sie geeignete Schutzmaßnahmen für Hardware, Software und lokale sowie verteilte Systeme konzipieren und einsetzen. Mit dem in der Vorlesung vermittelten Grundwissen können sie außerdem ihr Wissen über neue Bedrohungen fortlaufend aktualisieren und erweitern.

Kompetenzen:

LE#	Lernergebnis (LE)	Geprüft durch
LE1	Die Sicherheitslage in IT-Systemen und im Internet realistisch einzuschätzen.	Praktikum, Klausur
LE2	Die typischen Schwachstellen von Informationstechnischen Systemen zu kennen.	Praktikum, Klausur
LE3	Die Ansätze von Angriffen und Schadsoftware zu verstehen.	Praktikum, Klausur
LE4	Die Grundmechanismen für die Absicherung informationstechnischer Systeme zu kennen und zu verstehen.	Praktikum, Klausur
LE5	Die Umsetzung dieser Grundmechanismen in Informationssystemen und –netzen zu kennen.	Praktikum, Klausur
LE6	Angriffe und Sicherungsmaßnahmen exemplarisch durchzuführen und zu testen.	Praktikum
LE7	Für neu zu entwerfende Systeme geeignete Sicherheitsmechanismen auszuwählen.	Klausur

Inhalt:

In der Veranstaltung werden ausgewählte Kapitel von Stallings/Brown: Computer Security zugrunde gelegt. Es werden Grundbegriffe und Prinzipien der Sicherheit eingeführt. Dazu gehören Kryptografische Verfahren, Zugriffsverfahren, Identitätsmanagement u.a. Varianten von Schadsoftware und ihre Voraussetzungen werden behandelt. Werkzeuge zur Erkennung von Verwundbarkeiten und Angriffen werden dargestellt. Die Vorgehensweise zur Erstellung sicherer Software und für den sicheren Betrieb einer IT-Landschaft wird aufgezeigt. Das Thema Computerforensik wird ebenfalls exemplarisch behandelt. Im Praktikum werden auf Basis einer isolierten Infrastruktur Versuche zu den Inhalten der Vorlesung durchgeführt. Als Grundmuster dient das didaktische Prinzip von Angriff und Verteidigung, soweit anwendbar.

Medienformen:

Seminaristische Vorlesung, Folien, Tafel.
 Praktikum im Labor IT-Management mit praktischen Übungen in Teams, Vortrag der Ergebnisse durch Studierende.

Literatur:

- Eckert, Claudia (2013): IT-Sicherheit. Konzepte - Verfahren - Protokolle. 8., aktualisierte und korr. Aufl. München: Oldenbourg.
- Stallings, William; Brown, Lawrie (2012): Computer security. Principles and practice. 2. ed., international ed. Boston, Mass.: Pearson (Always learning).
- Bundesamt für Sicherheit in der Informationstechnik: Grundschutzhandbuch der IT Sicherheit <http://www.bsi.bund.de/gshb/index.htm>
- Aktueller IBM X-Force Jahresbericht; <http://www.ibm.com/services>

Modul:	Computergrafik	
Kürzel:	mkiB46	
Untertitel:		
Lehrveranstaltungen:	Vorlesung Praktikum	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Dr. Uwe Kloos	
Dozent(in):	Prof. Dr. Uwe Kloos	
Sprache:	Deutsch, Englisch (bei Bedarf)	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 4. Semester	
Lehrform/SWS:	Vorlesung, Praktikum	4 (2+2) SWS
Arbeitsaufwand:	Präsenzstudium	60 Stunden
	Eigenstudium	90 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung:	mkiB31, mkiB32, Programmieren in C++	
Studien-/Prüfungsleistungen/ Prüfungsform:	Vorlesung: Klausur Praktikum: Praktikum	

Modulziel

Das Ziel des Moduls ist es, die Studierenden für das Themengebiet Computergrafik zu sensibilisieren und sie in die Lage zu versetzen, 3D-Grafikprogramme zu entwickeln und zu verstehen. Als Erweiterung zu den medialen Modulen Grafik, Audio und Video werden in diesem Modul computerbasierte 3D-Grafikverfahren behandelt, wobei die Kompetenzen aus den anderen medialen Modulen eingesetzt werden können. Im weiteren Verlauf des Studiums soll mit einem erfolgreichen Bestehen sichergestellt sein, dass die Studierenden in der Lage sind grafische Programme zu entwickeln sowie grafische Anwendungen verstehen und bedienen zu können.

Angestrebte Lernergebnisse:

Kenntnisse:

- Die Studierenden kennen die mathematische Basis zur Berechnung von Transformationen dreidimensionaler Objekte sowie verschiedener Vektoroperationen.

- Die Studierenden kennen die Grafikpipeline und die verschiedenen Verfahren, die innerhalb dieser Pipeline genutzt werden.
- Die Studierenden sind in der Lage lokale und globale Beleuchtungsverfahren zu benennen und kennen deren Unterschiede.
- Die Studierenden kennen verschiedenen Texturierungsverfahren und sind in der Lage diese auf 3D-Objekte anzuwenden.
- Die Studierenden kennen unterschiedliche Methoden der Modellierung und sind in der Lage aus einfachen Objekten komplexe 3D-Welten aufzubauen.
- Die Studierenden können verschiedene Formate zum Austausch von Objekten in 3D-Welten benennen und kennen deren Vor- und Nachteile.
- Die Studierenden kennen den Unterschied zwischen statischen Bildern und dynamischen Animationen und sind in der Lage verschiedene Steuermechanismen zur Erzeugung einer Animation zu benennen und einzusetzen.

Fertigkeiten:

Die Studierenden berechnen eigenständig Transformationen und können die in der Computergrafik eingesetzten mathematischen Methoden bewerten und daraus eigene Verfahren konstruieren. Sie sind weiterhin in der Lage eine vorgegebene Aufgabenstellung aus der Computergrafik zu analysieren und vorhandene Algorithmen so zu bewerten, dass sie die passenden Verfahren zur effizienten Lösung der Aufgabenstellung herausuchen können. Die entwickelten Lösungen sind dabei auch in Hinblick auf Performanz konzipiert, wobei die Studierenden Grafikanwendungen auch diesbezüglich analysieren und bewerten können. Weiterhin sind sie in der Lage mit einem gängigen Grafikwerkzeug arbeiten zu können und einfache Animationen zu erstellen.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage:

LE#	Lernergebnis (LE)	Geprüft durch
LE1	Transformationen von 3D-Objekten in homogenen Koordinaten und Vektoroperationen manuell zu berechnen.	Klausur, Artefakt
LE2	Eine frei verfügbare Grafikbibliothek zu verstehen und produktiv einzusetzen.	Artefakt
LE3	Die erworbenen Kenntnisse eigenverantwortlich in einer Implementierung einer grafischen Anwendung umsetzen zu können.	Klausur, Artefakt
LE4	Die in Grafikanwendungen eingesetzten Verfahren zu analysieren und deren Vor- und Nachteile zu bewerten.	Klausur
LE5	Professionelle Modellierungs- und Animationswerkzeuge für die Erstellung von 3D-Modellen und Animationen einsetzen zu können.	Artefakt
LE6	Probleme und Grenzen, die bei der Entwicklung von grafischen Anwendungen entstehen, einzuschätzen.	Artefakt
LE7	Eigene Entwicklungen und Fähigkeiten beurteilen zu können.	Artefakt
LE8	Aktuelle Entwicklungen der Disziplin Computergrafik zu beurteilen und sich aneignen zu können.	Artefakt

Inhalt:

In der Vorlesung werden die Studierenden in das Themengebiet der Computergrafik und verwandte Gebiete (Animation, Visualisierung) herangeführt. Dazu gibt es jeweils einen theoretischen Block, in dem ausgewählte Themen detailliert behandelt und besprochen werden. Dabei steht die Vermittlung gängiger Algorithmen und Prinzipien im Vordergrund. Das theoretische Wissen wird dann in kleinen Übungseinheiten praktisch umgesetzt und basierend auf einer standardisierten Grafikbibliothek werden eigene Grafikprogramme entwickelt. Als Themen werden die grundlegenden Techniken und Verfahren vom Modell zum Bild behandelt (Transformationen, Projektionen, Sichtbarkeitstest, Farbgebung, Rasterisierung). Weiterhin stehen Themen wie Modellierung, lokale und globale Beleuchtungsrechnung, Texturen auf dem Plan. Bei der praktischen Umsetzung wird darauf geachtet, dass in der Industrie genutzte Werkzeuge eingesetzt werden, so dass auch ein praktisches Wissen erworben wird.

Medienformen:

Das Lehrmaterial besteht aus einem Folienskript, das in elektronischer Form verteilt wird bzw. über einen Zentralserver verfügbar ist und einer Einführung in OpenGL. Das Modul umfasst eine Vorlesung mit einem begleitenden Praktikum. Seminaristischer Unterricht, bei dem Beispiele zu den theoretischen Inhalten multimedial veranschaulicht werden. Die Studierenden bearbeiten individuell oder in Gruppen verschieden Übungsaufgaben aus dem Bereich der Computergrafik. Zunächst werden mathematische Verfahren, die die Basis der Transformationen bilden, in praktischen Übungen vertieft. Dann wird über mehrere Übungseinheiten hinweg schrittweise eine komplexe grafische Anwendung basierend auf einer frei verfügbaren Grafikbibliothek entwickelt. Zusätzlich werden grundlegende Verfahren der Modellierungs- und Animationstechnik besprochen und an einem praktischen Projekt umgesetzt. Bei der praktischen Umsetzung wird darauf geachtet, dass in der Industrie genutzte Werkzeuge eingesetzt werden. Die Betreuung bei den Programmierübungen und dem Animationsprojekt erfolgt durch den Dozenten.

Literatur:

- Angel, Edward (2006): Interactive computer graphics. A top-down approach using OpenGL. 4. ed., internat. ed. Boston, Mass., Munich: Pearson Addison-Wesley.
- Bender, Michael; Brill, Manfred (2006): Computergrafik. Ein anwendungsorientiertes Lehrbuch. 2., überarb. Aufl. München, Wien: Hanser.
- Nischwitz, Alfred; Haberäcker, Peter (2004): Masterkurs Computergrafik und Bildverarbeitung. Alles für Studium und Praxis; Bildverarbeitungswerkzeuge Beispiel-Software und interaktive Vorlesungen online verfügbar. 1. Aufl. Wiesbaden: Vieweg.
- Watt, Alan H. (2005): 3D computer graphics. 3. ed. [,repr.]. Harlow: Addison-Wesley.
- Weitere vertiefende Literatur wird jeweils in den Veranstaltungen bekannt gegeben.

Modul:	Praktisches Studiensemester	
Kürzel:	mkiB51	
Untertitel:		
Lehrveranstaltungen:	Praxissemester	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Dr. Uwe Kloos	
Dozent(in):	Professoren des Studienganges, Industriebetreuer	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 5. Semester	
Lehrform/SWS:		
Arbeitsaufwand:	Eigenstudium	900 Stunden
Kreditpunkte:	30 ECTS	
Voraussetzungen nach StuPro:	Alle Module aus Semester 1 – 3 müssen bestanden sein	
Empfohlene Voraussetzung:	Alle Module aus den Semestern 1 - 4	
Studien-/Prüfungsleistungen/ Prüfungsform:	Praxis: Praktikumsbericht	

Modulziele:

Das praktische Studiensemester stellt das im Studium erworbene Wissen in den Kontext projektbezogener fachlicher Tätigkeiten. Ziel des Moduls ist es facheinschlägige praktische Erfahrungen und soziale Kompetenzen in einer geeigneten Institution zu erwerben. Die Studierenden sollen dabei in der Rolle eines Bachelorabsolventen unter Anleitung eines akademisch qualifizierten Betreuers tätig sein. Das Modulziel ist die Vermittlung praktischer Vorgehensweisen zu Studieninhalten unter den organisatorischen Gegebenheiten von Projekten.

Angestrebte Lernergebnisse:

Kenntnisse:

Die Studierenden kennen nach dem praktischen Studiensemester facheinschlägige Methoden des Projektmanagements aus der Praxis. Sie lernen im praktischen Studiensemester die methodischen und fachpraktischen Ausprägungen der Studieninhalte

kennen. Sie erwerben Kenntnisse in der Umsetzung der Studieninhalte mit begrenzten Ressourcen im Team unter vorgegebenen wirtschaftlichen und rechtlichen Rahmenbedingungen.

Fertigkeiten:

Die Studierenden haben die praktischen und kognitiven Fertigkeiten zu den fachlichen Inhalten des Studiums weiterentwickelt. Sie haben die praktischen Fertigkeiten zur Handhabung der Verfahren und Methoden des Studiums ausgeweitet. Ihre Fertigkeiten in der Bearbeitung von anwendungsspezifischen Fragestellungen mit facheinschlägigen Methoden im idealerweise multidisziplinären Team wurden ausgeweitet.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage, die im Studium erworbenen Kenntnisse und Fähigkeiten in der Praxis unter den organisatorischen, wirtschaftlichen und rechtlichen Rahmenbedingungen eines realen Projektes umzusetzen.

Die erworbenen fachlichen Kompetenzen werden durch einen Bericht mit einer wissenschaftlichen Ausarbeitung, in der die inhaltlichen Arbeits- und Projektbeiträge, das erworbene Wissen und die Vorgehensweisen erörtert werden, nachgewiesen.

Dem Bericht sind der kalendarischer Tätigkeitsnachweis und das Praktikantenzeugnis in dem die Präsenzzeit ausgewiesen wird beizufügen. Die Institution in der das praktische Studiensemester absolviert wurde bescheinigt durch Unterschrift die Präsenzzeit, die inhaltliche Richtigkeit der kalendarischen Tätigkeitsbeschreibung und des fachlichen Berichtes.

Inhalt:

Die konkreten, den Studierenden übertragenen Aufgaben und die vermittelten Kenntnisse und Fähigkeiten können entsprechend der Unternehmenspraxis unterschiedlich sein. Wichtig ist, dass die Studierenden exemplarische Einsichten im Rahmen des Studienziels gewinnen können, und so die Theorie/Praxis-Beziehung des Medien- und Kommunikationsinformatikstudiums unterstützt wird. Idealerweise arbeiten die Studierenden bei der Planung, Analyse, Konzeption, Entwicklung, dem Betrieb oder der Anwendung von Software-, Medien- und Kommunikationssystemen in einem Projekt aktiv mit.

Praktikantenstellen müssen vor dem Antritt des Praktikums von der Fakultät (Praktikantenamt) genehmigt werden. Kriterien sind die Dauer, insbesondere die vorhandene Betreuung durch eine akademisch qualifizierte Person und die Aufgabenstellungen die konform mit den Ausbildungsinhalten des Studiengangs Medien- und Kommunikationsinformatik sind. Die Studierenden sollen die Tätigkeiten eines Bachelor in Medien- und Kommunikationsinformatik unter Anleitung der betreuenden Person verrichten. Dies muss aus dem Bericht ersichtlich sein.

Medienformen:

Der Bericht, der kalendarische Tätigkeitsnachweis und das Praktikantenzeugnis werden elektronisch abgegeben.

Literatur:

Web Seiten des Praktikantenamtes mit Detaillierten Hinweisen

Modul:	Wahlpflicht 1 und 2
Kürzel:	mkiB61, mkiB71
Untertitel:	
Lehrveranstaltungen:	<p>mkiB61, benotet:</p> <p>mkiBW101 Medien- und Kommunikationsinformatik Projekt 1 mkiBW102 Medizininformatik mkiBW103 Multimodale Signalverarbeitung mkiBW104 Eingebettete Systeme und Robotik mkiBW105 E-Health mkiBW106 Medizinische Informationssysteme mkiBW107 Medizinische Visualisierung und Simulation mkiBW108 Grundlagen der Wirtschaftswissenschaften mkiBW109 Grundlagen in Marketing und Unternehmenskommunikation mkiBW110 Robotersysteme mkiB71, unbenotet:</p> <p>mkiBW201 Medien- und Kommunikationsinformatik Projekt 2 mkiBW202 Medizinische Grundlagen mkiBW203 Standards und Prozesse der Medizinisch- Technischen Informatik mkiBW204 Einführung in Statistik und Biometrie mkiBW205 Qualitätsmanagement im Gesundheitswesen mkiBW206 Grundlagen und Methoden der Wirtschaftsinformatik mkiBW207 Logistik und Produktion mkiBW208 Unternehmensmodellierung mkiBW209 Management und Controlling mkiBW210 System und Anwendungsbetrieb mkiBW211 Development of Smart Textiles mkiBW212 Data Mining</p>
Studiensemester:	jedes Semester
Modulverantwortlicher:	Prof. Dr. Peter Hertkorn
Dozent(in):	Dozenten der gewählten Lehrveranstaltungen
Sprache:	Deutsch, Englisch
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 6. und 7. Semester

Lehrform/SWS:	Vorlesung	4 SWS
	mkiBW101 mki-Projekt 1	2 SWS
	mkiBW201 mki-Projekt 2	2 SWS
Arbeitsaufwand:	Präsenzstudium	60 bzw. 30 Stunden
	Eigenstudium	90 bzw. 120 Stunden
Kreditpunkte:	je 5 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung:	abhängig von der gewählten Veranstaltung	
Studien-/Prüfungsleistungen/ Prüfungsform:	abhängig von der gewählten Veranstaltung	

Modulziele:

Die Studierenden ergänzen das bisher erlernte Fachwissen der Medien- und Kommunikationsinformatik durch Inhalte angrenzender Fachdisziplinen. Dadurch ist es möglich, das Studium nach persönlichen und beruflichen Zielen anzupassen und eine individuelle Schwerpunktsetzung vorzunehmen. Der Katalog der angebotenen Wahlpflichtfächer kann in Einzelfällen in Absprache mit dem Prüfungsausschuss erweitert werden, wenn das der individuellen Profilbildung eines Studierenden Rechnung trägt

Angestrebte Lernergebnisse:

Kenntnisse/ Fertigkeiten/ Kompetenzen:

Die Studierenden erlernen neben den fachlichen Inhalten auch die Sprache und Kultur eines angrenzenden Fachgebiets kennen.

Die Studierenden lernen neben den fachlichen Fertigkeiten ihr bisher erworbenes Wissen in einem neuen Kontext einzubringen und Wissen aus bisher unbekanntem Gebieten in ihr Portfolio zu integrieren.

Inhalt:

Abhängig von der gewählten Veranstaltung.

Medienformen:

Abhängig von der gewählten Veranstaltung.

Literatur:

Abhängig von der gewählten Veranstaltung.

Modul:	Mobile Computing	
Kürzel:	mkiB62	
Untertitel:		
Lehrveranstaltungen:	Vorlesung Praktikum	
Studiensemester:	jedes Semester	
Modulverantwortliche:	Prof. Dr. Natividad Martínez	
Dozent(in):	Prof. Dr. Natividad Martínez	
Sprache:	Deutsch, Englisch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 6. Semester	
Lehrform/SWS:	Vorlesung, Praktikum	4 (2+2) SWS
Arbeitsaufwand:	Präsenzstudium	60 Stunden
	Eigenstudium	90 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	Keine	
Empfohlene Voraussetzung :	mkiB35, mkiB44, mkiB45	
Studien-/ Prüfungsleistungen/ Prüfungsform:	Vorlesung: Continuous Assessment	

Modulziele:

Die Veranstaltung Mobile Computing führt in die Thematik verteilter und mobiler Systeme ein. Ziel des Moduls ist es, dass die Studierenden die Prinzipien mobiler Kommunikationsnetze kennen und die wichtigsten Technologien zur Entwicklung mobiler Anwendungen benutzen können. Die Vorlesung baut auf die erworbenen Kenntnisse über Webprogrammierung (Vorlesung Verteilte Systeme) und Rechnernetze (Vorlesung Internetworking) auf und bezieht sich ebenfalls auf IT-Sicherheitsaspekte.

Angestrebte Lernergebnisse:

Kenntnisse:

Die Studierenden erwerben Kenntnisse über die Funktionsweise von Protokollen für mobile Anwendungen auf Basis des Internets. Sie lernen, welche Anforderungen diese

Anwendungen stellen und wie diese von Protokollen und Diensten auf tieferen Ebenen erfüllt werden.

Darüber hinaus erlernen die Studierenden Grundlagenwissen für Mobile Computing und lernen dabei typische Szenarien und Aufgabenstellungen kennen. Sie sind in der Lage, Ausführungsplattformen zu beschreiben und lernen unterstützende Technologien kennen. Die Studierenden kennen die Grundlagen des Internet der Dinge und die Haupteigenschaften der Sensoren und Sensornetze.

Fertigkeiten:

Die Studierenden wenden die erworbenen Kenntnisse über mobile Kommunikationen selbstständig beim Lösen von Übungsaufgaben an. Sie lernen den Einsatz von hardware- und softwarebasierten Analysewerkzeugen und verfolgen Kommunikationsprozesse auf Protokollebene.

Die Studierende vergleichen unterschiedliche mobile Betriebssysteme und Ausführungsplattformen unter bestimmten Anforderungen. Sie programmieren Problemlösungen auf typischen Plattformen (z.B. Smartphones, TabletPCs, eingebettete Systeme etc.). Darüber hinaus planen die Studierenden die Integration von Sensoren und Sensornetzen und erarbeiten (umgebungsabhängige) Anwendungen.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage:

LE#	Lernergebnis (LE)	Geprüft durch
LE1	Unterschiedliche mobile Technologien zu analysieren und zu beurteilen.	Klausur
LE2	Lösungsvorschläge für Beispielszenarien auf der Basis der gelernten mobilen Technologien zu erarbeiten.	Übungen
LE3	Eine mobile Anwendung unter bestimmten Anforderungen und Randbedingungen zu entwerfen.	Projektarbeit
LE4	Im Team zu arbeiten um komplexe Aufgaben zu lösen.	Projektarbeit
LE5	Moderne Entwicklungsumgebungen und Werkzeuge zu benutzen.	Projektarbeit
LE6	Selbständig Information über aktuelle Entwicklungen der Disziplin Mobile Computing zu suchen, zu beurteilen und wissenschaftlich zu kommunizieren.	Ausarbeitung
LE7	Themen der Disziplin Mobile Computing unter Verwendung der Fachsprache kompetent zu präsentieren und zu diskutieren.	Referat

Inhalt:

Die Vorlesung Mobile Computing gliedert sich in drei Hauptgebiete: Mobile Kommunikation, Programmierung mobiler Anwendungen und Internet der Dinge. Dabei wird der Inhalt sehr umfangreich und anwendungsorientiert diskutiert. Deswegen wurde die Prüfungsform „Continuous Assessment“ ausgewählt, um die Studierenden mit unterschiedlichen Artefakten begleiten zu können. Zu jedem Gebiet gehört eine theoretische Einführung, deren Verständnis mit Testaten geprüft wird (LE1), dazu gehören Übungen (LE2), die die Studierenden allein oder in Gruppen lösen. Anschließend gibt es praktische Laboraufgaben (LE5), die in Gruppen gelöst werden und die zu einer Projektarbeit gebündelt werden (LE3,

LE4). In der zweiten Hälfte der Vorlesung bereiten die Studierenden eine Ausarbeitung über ein aktuelles Thema der Disziplin vor (LE6) und tragen Ihre Erkenntnisse vor (LE7). Die Gebiete sind:

1. Grundsätze der Mobilien Kommunikation: drahtlose und mobile Netzwerke [Schiller].
2. Programmierung mobiler Anwendungen: Einführung in Web-Apps-Programmierung mit HTML5 und native Apps mit Android, iOS, Windows-Phone und QNX. Vertiefung der Programmierung in Android [Meier].
3. Einführung in das Internet der Dinge [Kalin] [Ewen].

Medienformen:

Abhängig von Inhalt und gezielter Kompetenz werden unterschiedliche Medienformen ausgewählt. Einige Themen werden klassisch mit Folienskripten behandelt, die mit dem Beamer projiziert und durch den Einsatz der Tafel vertieft, erklärt und veranschaulicht werden. In einem seminaristischen Unterrichtstil werden die Studierenden einzeln oder in Gruppen Themen mit Hilfe von ausgesuchten Literaturreferenzen bearbeiten und vortragen. Das Modul umfasst eine Vorlesung mit begleitendem Praktikum. Dafür werden die Studierenden Hinweise über die notwendige Installation und die Anforderungen an die Systeme erhalten, die sie prototypisch unter Betreuung der Dozenten im Labor entwickeln sollten.

Literatur:

- McEwen, Adrian; Cassimally, Hakim (2013): Designing the Internet of Things. Online-Ausg. Hoboken: Wiley (EBL-Schweitzer). Online verfügbar unter <http://swb.ebib.com/patron/FullRecord.aspx?p=1471865>.
- Meier, Reto (2012): Professional Android 4 application development. Indianapolis, Ind.: Wiley (Programmer to programmer).
- Poslad, Stefan (2009): Ubiquitous computing. Smart devices environments and interactions. 1. publ. Chichester: Wiley.
- Schiller, Jochen H. (2003): Mobile communications. 2nd ed. London: Addison-Wesley.
- Darüber hinaus aktuelle Artikel aus Fachjournalen und Konferenzen sowie Internet Ressourcen.

Modul:	Recht & BWL	
Kürzel:	mkiB63	
Untertitel:		
Lehrveranstaltungen:	Vorlesung	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Boris Terpinc	
Dozent(in):	Frau Regina Brauchler, Prof. Manfred Gerblinger	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 6. Semester	
Lehrform/SWS:	Vorlesung	4 SWS
Arbeitsaufwand:	Präsenzstudium	60 Stunden
	Eigenstudium	90 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung:	keine	
Studien-/Prüfungsleistungen/ Prüfungsform:	Vorlesung: Hausarbeit, Referat	

Modulziele:

Betriebswirtschaftliches Management beschäftigt sich im weiteren Sinne mit dem Analysieren, Bewerten und Gestalten von Betrieben und Geschäftsprozessen. Ausgehend von der Beziehung zwischen Medienunternehmen und Markt werden die Ziele und Instrumente der strategischen Planung in Medienunternehmen erläutert.

Die Studierenden lernen gemäß dem Train-the-Trainer-Ansatz die theoretischen Geschäftsprozessmodelle und Instrumente der betriebswirtschaftlichen Unternehmensorganisation kennen und erproben im Rahmen der Veranstaltung direkt die Anwendung im Gedanken an die Abteilung, in der sie im Praktikum oder in der Ausbildung bereits tätig waren.

Der rechtliche Teil beschäftigt sich zunächst mit einem Überblick über die Rechtssystematik und einer Einführung in zentrale Grundlagen der Rechtsordnung: Begriffe, Funktionen und Erscheinungsformen des Rechts, Rechtsanwendung und –Durchsetzung, Einblick in juristische Methoden. Im Hauptteil werden unterschiedliche rechtliche Aspekte der Medien- und Kommunikationsinformatik behandelt. Die Studierenden sollen einen Überblick über die

zentralen Rechtsprobleme der Medien- und Kommunikationsinformatik erhalten. Sie werden dadurch in die Lage versetzt, die Problempunkte zu erkennen und durch entsprechende vertragliche Gestaltung Vorsorge zu treffen.

Angestrebte Lernergebnisse:

Kenntnisse:

- Kennen von Informationen, Theorie-und/oder Faktenwissen zu aufbau- und ablauforganisatorischen Maßnahmen im Betrieb.
- Kennen von studienfachrelevanten rechtlichen Grundlagen.
- In Recherche und in der Entwicklung von Filmideen und der Umsetzung im Medium Video.
- Der Produktionsschritte in der Filmherstellung.
- Der Regeln der Teamarbeit, Arbeitsteilung und Teamleitung.
- Der audiovisuellen Gestaltungsmittel.
- Der arbeits- und sicherheitstechnischen Richtlinien und Maßnahmen.
- Der Grundbegriffe journalistischen Schreibens.

Fertigkeiten:

Kognitive und praktische Fertigkeiten im Berechnen von Kennzahlen des betrieblichen Managements zur Entscheidungsvorbereitung. Dokumentation von Organisationseinheiten. Prozess- und Ablaufdokumentation, -steuerung – und Optimierung mit Folgestruktur- und Netzplantechnik. Durchführen von Aufgabenanalyse, -synthese und Kritik. Vermittlung von fachübergreifenden Kompetenzen, speziell die Berücksichtigung von rechtlichen Rahmenbedingungen in Verbindung mit IT-Systemen.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage:

LE#	Lernergebnis (LE)	Geprüft durch
LE1	Kleinere Studien in Betrieben durchzuführen, um die Ergebnisse als Dokumentationsgrundlage für eine betriebliche Projekt- oder Bachelorarbeit zu verwenden.	Hausarbeit
LE2	Stellenbeschreibungen, Funktionsplänen und Organigrammen im Hinblick auf die effektive und effiziente Umsetzung eines aufgabenbezogenen Vorschlagswesens im Betrieb zu erstellen.	Hausarbeit
LE3	Ablaufdiagrammen zur Prozesssteuerung und -optimierung mit der Analyse von Bearbeitungs- und Durchlaufzeiten auszuarbeiten.	Hausarbeit
LE4	Kapazitätsplanung zu analysieren und Auslastungs- und Nutzungsgraden für die Arbeitsvorbereitung der Produktionsplanung zu ermitteln.	Hausarbeit
LE5	Rechtliche Aspekte des nachgefragten Berufs zu beurteilen und einzuschätzen.	Referat

Inhalt:

Teil BWL:

In Kapitel 1 wird das ganzheitliche Organisationskonzept mit Strategien und Zielen betrachtet, wie es sich mit aufbau- und ablauforganisatorischen Maßnahmen anhand entsprechender Kenngrößen des Betriebes, analysieren, dokumentieren und gestalten lässt. Im Vordergrund stehen Aufgaben, Kapazitäten, Auslastungs- und Nutzungsgrade sowie Bearbeitungs- und Durchlaufzeiten. Im Anschluss werden verschiedene Geschäftsprozessmodelle aufgezeigt. Der Grundgedanke der Prozessorientierung wird erläutert und auf verschiedene Geschäftsebenen übertragen. Weiterhin wird betriebswirtschaftliches Handeln aus Sicht der Informationstechnologie im Rahmen der Prozess-Strukturierung mit dem Erstellen von Lasten- und Pflichtenheft sowie Anforderungskatalog und Evaluationsbogen dargestellt. Das betriebliche Rechnungswesen führt in Kapitel 2 zur Ermittlung des Betriebserfolges und zur Bilanzierung. In Kapitel 3 steht die Selbstkostenermittlung auf der Basis der Betriebsabrechnung im Rahmen der Kosten- und Leistungsrechnung im Mittelpunkt.

Abschließend werden Kennzahlen wie Wirtschaftlichkeit, Amortisation und Rentabilität ermittelt sowie die Besonderheiten des Medien-Managements auch unter dem Total Quality Management-Ansatz herausgearbeitet.

Teil Recht:

IT-Vertragsrecht (Hard- und Software):

Überblick, Einordnung in das Rechtssystem, vertragsrechtliche Gestaltungsmöglichkeiten, Hard- und Softwareüberlassungsverträge, Mängelhaftung (auch Produkthaftung), Software-, Vertrags- und Lizenzrecht.

Arbeitsrechtliche Fragestellungen:

Persönliche Haftung von verantwortlichen Funktionsträgern eines Unternehmens im Falle von Urheberrechtverletzungen in dem von ihnen zu verantwortenden Bereich Strafrecht: u.a. Computerdelikte.

Urheberrecht:

Schutz von 'geistigem Eigentum', Rechtsschutz und Verwertung von Computerprogrammen, Rechtsschutz für Informationssysteme und Datenbanken, Rechtsfragen der Open Source Software, einzelne Lizenztypen und das Patentrecht und Markenrecht: u.a. der markenzeichenrechtliche Schutz von Computerprogrammen.

Wettbewerbsrecht und Abmahnung, Kauf im Internet.

Im Internetrecht:

Domainrecht, Impressum u.a. Datenschutzrecht und neuere Rechtsentwicklungen, EU-Richtlinien und ihre Transformation in nationales Recht.

Medienformen:

Geschäftsprozessmanagement, das Arbeitsanalyse- und Synthesekonzept sowie die Aufgabenkritik werden ebenso wie die Instrumente der Prozessstrukturierung und -gestaltung in Form von Begleittexten mit Übungen sowie mit Präsentationen via Laptop und Beamer im Wechsel dargeboten, wobei auf Präsentationszeiten von 20 Minuten zur Wissensvermittlung in der Regel kurze Anwendungsübungen mit Beispielen folgen, die in der Lerngruppe zu lösen sind. Nach dieser vertiefenden Bearbeitung an Praxisbeispielen werden die Ergebnisse zusammengefasst und auf die betriebliche Praxis der Studierenden übertragen. Videos aus der Automobilbranche werden ebenso wie Fotos aus dem Verlagsgeschäft zur Verdeutlichung heran gezogen.

Literatur:

Teil BWL:

- Binner, Hartmut F. (2004): Handbuch der prozessorientierten Arbeitsorganisation. Methoden und Werkzeuge zur Umsetzung. 1. Aufl. München, Wien: Hanser (REFA-Fachbuchreihe Unternehmensentwicklung).
- Karmasin, Matthias; Winter, Carsten (2000): Grundlagen des Medienmanagements. München: Fink (UTB, 8203).
- Wöhe, Günter (1981): Einführung in die allgemeine Betriebswirtschaftslehre. 14., überarb. Aufl. München: Vahlen (Vahlens Handbücher der Wirtschafts- und Sozialwissenschaften).

Teil Recht:

- Eigenes Skript, Übersichtblätter und Arbeitsblätter mit Verständnisfragen.
- Dreier, Thomas Dreier, Thomas, Skript Internetrecht, Online-Version, Juni 2009: Verfügbar unter <http://www.zar.uni-karlsruhe.de/653.php>.
- Hoeren, Thomas Hoeren, Thomas, Internetrecht, Online-Version, Oktober 2012. Verfügbar unter: <http://www.uni-muenster.de/Jura.itm/hoeren/lehre/materialien>.

Modul:	Seminar ausgewählte Themen der Informatik	
Kürzel:	mkiB64	
Untertitel:	Einführung in das wissenschaftliche Arbeiten	
Lehrveranstaltungen:	Seminar	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Dr. Christian Kücherer	
Dozent(in):	Prof. Dr. Christian Kücherer	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 6. Semester	
Lehrform/SWS:	Vorlesung	2 SWS
Arbeitsaufwand:	Präsenzstudium	30 Stunden
	Eigenstudium	120 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:		
Empfohlene Voraussetzung:	alle Module der ersten 4 Semester	
Studien-/Prüfungsleistungen/ Prüfungsform:	Seminar: Hausarbeit, Referat	

Modulziele:

Das „Seminar ausgewählte Themen der Informatik“ führt den Studierenden an die Bearbeitung der Bachelor-Thesis und an das Bachelor-Kolloquium heran.

Wie in der Bachelor-Thesis, verlangt dieses Seminar eine fundierte Literatur-Recherche und Auswertung in einem Themengebiet der Informatik. Auf Basis der Literatur und auf Basis des Wissens des Studierenden aus den ersten drei Semestern ist eine schriftliche Ausarbeitung nach formalen Kriterien anzufertigen. Dabei muss eine eigene Argumentationslinie erkennbar sein, die sich auf der Literaturarbeit stützt.

Unter ähnlichen Bedingungen wie im Bachelor-Kolloquium oder wie in einem Konferenzvortrag ist diese Argumentationslinie im Rahmen einer Powerpoint-Präsentation zu verteidigen.

Angestrebte Lernergebnisse:

Kenntnisse:

- Die Idee und Vorgehensweise des wissenschaftlichen Arbeitens zu kennen.

- Die Arten der Fragestellungen zu einer wissenschaftlichen Arbeit (Beschreibung, Erklärung, Prognose, Gestaltung, Kritik/Bewertung) benennen und erklären zu können.
- Zitierbare Quellen nennen und klassifizieren zu können.
- Literatur als Primärliteratur, Sekundärliteratur und Tertiärliteratur einordnen zu können.
- Die Bedeutung von Abduktion, Deduktion und Induktion zu kennen.
- Die Technik des direkten und indirekten Zitierens zu kennen.
- Die verschiedenen Gliederungsarten einer wissenschaftlichen Arbeit nennen und beschreiben zu können.

Fertigkeiten:

- Literatur zu sichten, deren Güte zu bewerten und für die eigene Argumentation nutzbar machen zu können.
- Literatur zu bewerten, korrekt zu zitieren und in die eigene Argumentation aufnehmen zu können.
- Eine eigene Argumentationslinie aufzubauen und das Für und Wider einer Entscheidung zu diskutieren.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage:

LE#	Lernergebnis (LE)	Geprüft durch
LE1	Eine Literaturrecherche nach wissenschaftlichen Standards durchzuführen und auf der Basis von Literatur argumentieren zu können.	Artefakt
LE2	Eine schriftliche Ausarbeitung nach wissenschaftlichen Standards auf der Basis einer Literaturarbeit mit einer eigenen Argumentationslinie unter zeitlicher Beschränkung anzufertigen.	Artefakt
LE3	Die eigene Argumentationslinie vorzustellen, diskutieren und verteidigen zu können.	Präsentation

Inhalt:

Das Seminar beginnt mit einer Blockvorlesung zum Thema „wissenschaftliches Arbeiten“. Nach der Definition von wissenschaftlichem Arbeiten werden Strategien zur Themenfindung anhand von vordefinierten Fragen [Karmasin, Ribing 2007] vorgestellt. Im Anschluss daran werden Methoden zur iterativ-inkrementellen Literatur-Recherche [Distrer 2011, Kitchenham und Charters 2007] mit Kriterien zur Güte der verwendeten Literatur vorgestellt. Um wissenschaftlich argumentieren zu können, werden die Abduktion, Deduktion und Induktion als Strategien des Schlussfolgerns [Karmasin, Ribing 2007] besprochen. Um Literaturangaben korrekt wiederzugeben, werden die Arten des Zitierens, typische Plagiate, die Anfertigung von Literaturverzeichnissen und die Grundlagen guter wissenschaftlicher Praxis vorgestellt. Zum Abschluss der Blockvorlesung werden verschiedene Arten der Gliederung einer wissenschaftlichen Ausarbeitung präsentiert [Esselborn-Krumbiegel 2002] (LE1). Die Studierenden wählen daraufhin ein Thema zur Bearbeitung aus einem Themenpool. Dazu ist eine schriftliche, wissenschaftlich fundierte Ausarbeitung nach bestimmten Kriterien anzufertigen, die einem Konferenzbeitrag ähneln und die strikt einzuhalten sind. Eine Erstversion wird nach ca. 8 Wochen vom Dozenten einem Review unterzogen. Eine finale Version der Ausarbeitung ist dann ca. nach weiteren 4 Wochen anzufertigen. Für die Abgaben der ersten und der finalen Version gelten strenge Deadlines (LE2).

Abschließend ist das Thema dann in einem Kurzvortrag vorzustellen und die Argumentationslinie zu verteidigen (LE3).

Medienformen:

- Geblockter seminaristischer Unterricht mit PC-Beamer und Foliensatz zur Einführung in das wissenschaftliche Arbeiten. Teile des Foliensatzes und die Regeln zum Ablauf der Veranstaltung stehen zu Beginn der Veranstaltung zum Download bereit.
- Beantwortung von Einzelfragen an den Dozenten per E-Mail und/oder während der Sprechstunde.
- Abgabe einer schriftlichen Ausarbeitung des Studierenden nach vorgegebenen Kriterien. Die Erstversion wird einem Review unterzogen. Die finale Version wird bewertet mit 60% der Gesamtnote.
- Verteidigung des Themas durch einen Vortrag als eigene Powerpoint-Präsentation des Studierenden mit anschließender Befragung aus dem Auditorium. Die Verteidigung wird bewertet mit 40% der Gesamtnote.

Literatur

- Brink, Alfred (2005): Anfertigung wissenschaftlicher Arbeiten. Ein prozessorientierter Leitfaden zur Erstellung von Bachelor- Master- und Diplomarbeiten in acht Lerneinheiten. 2., völlig überarb. Aufl. München, Wien: Oldenbourg.
- Disterer, Georg (2011): Studienarbeiten schreiben. Seminar- Bachelor- Master- und Diplomarbeiten in den Wirtschaftswissenschaften. 6., vollst. überarb. und erw. Aufl. Berlin, Heidelberg: Springer (Springer-Lehrbuch).
- Esselborn-Krumbiegel, Helga (2002): Von der Idee zum Text. Eine Anleitung zum wissenschaftlichen Schreiben im Studium. Paderborn, München, Wien, Zürich: Schöningh (UTB, 2334).
- Karmasin, Matthias; Ribing, Rainer (2007): Die Gestaltung wissenschaftlicher Arbeiten. Ein Leitfaden für Haus- und Seminararbeiten Magisterarbeiten Diplomarbeiten und Dissertationen. 2., aktualisierte Aufl. Wien: WUV Facultas-Verl (UTB, 2774 : Arbeitshilfen).
- Kitchenham, Barbara, Charters Stuart: Guidelines for performing systematic literature reviews in software engineering - Version 2.3. Keele, Staffs, UK; Durham, UK; 2007. Report No.: EBSE Technical Report 2007-001.
- Kruse, Otto (2004): Keine Angst vor dem leeren Blatt. Ohne Schreibblockaden durchs Studium. 10. Aufl. Frankfurt/Main: Campus-Verl (Campus concret, 16).
- Rechenberg, Peter (2003): Technisches Schreiben. (nicht nur) für Informatiker. 2., erw. Aufl. München, Wien: Hanser.
- Schneider, Wolf (2007): Deutsch! Das Handbuch für attraktive Texte. 2. Aufl. Reinbek bei Hamburg: Rowohlt-Taschenbuch-Verl (rororo, 61993 : Sachbuch).
- Theisen, Manuel René (2002): Wissenschaftliches Arbeiten. Technik - Methodik - Form. 11., aktualis. Aufl. München: Vahlen (WiSt-Taschenbücher).

Modul:	Cloud Computing	
Kürzel:	mkiB65	
Untertitel:		
Lehrveranstaltungen:	Vorlesung	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Dr. Marcus Schöller	
Dozent(in):	Prof. Dr. Marcus Schöller	
Sprache:	Englisch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 6. Semester	
Lehrform/SWS:	Vorlesung	4 SWS
Arbeitsaufwand:	Präsenzstudium	60 Stunden
	Eigenstudium	90 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung:	mkiB33, mkiB34, mkiB35, mkiB41, mkiB42, mkiB43, mkiB44, mkiB45	
Studien-/Prüfungsleistungen/ Prüfungsform:	Vorlesung: Klausur Praktikum: Praktikum	

Modulziele:

Mobile Computing und Cloud Computing bedingen heute einander. Die Teilnehmer sollen umfassende Kenntnisse bei der Gestaltung, Entwicklung und des Betriebs verteilter Anwendungen, fokussiert auf den Einsatz der grundlegenden Ausprägungen von Cloud-Diensten und ihrer Liefermodelle erwerben. Dazu sind die Kenntnisse der oben empfohlenen Voraussetzungen erforderlich. Die Veranstaltung Verteilte Systeme schafft die allgemeinen Voraussetzungen in Hinblick auf Webprogrammierung und Webservices.

Angestrebte Lernergebnisse:**Kenntnisse:**

Absolventen dieses Moduls verfügen nach dem erfolgreichen Abschluss über Kenntnisse über die Prinzipien und Charakteristiken von Cloud Computing. Sie können typische Dienste und Liefermodelle beschreiben und in Hinblick auf Fallbeispiele beurteilen. Sie haben ein Verständnis entwickelt für technische, organisatorische, kommerzielle, rechtliche, soziale

und sicherheitsrelevante Aspekte des Cloud Computing und kennen wesentliche Werkzeuge und Plattformen.

Fertigkeiten:

Die Studierenden können Anforderungen von Server-Diensten analysieren und geeignete Deployment-Varianten entwickeln und bewerten. Diese Varianten reichen von inhouse Server-Lösung über hybrid Cloud Modelle bis hin zu reinen Cloud-Lösungen. Dazu wenden sie eine Reihe von erlernten Methoden an. Basierend auf diesen Anforderungen können die Studenten Dienste entwickeln, die sich die Charakteristiken der Cloud zu Nutze machen. Des Weiteren kennen die Studierenden Details von Cloud-Umgebungen, was Ihnen einen tiefergehenden Vergleich der unterschiedlichen Deployment-Varianten ermöglicht. Auch die Anwendungsentwicklung für die Cloud wird in Workshops, z.B. IBM Cloud und Docker, eingeübt. Weitere wesentliche Werkzeuge, z.B. Chef und Puppet, Plattformen, z.B. Openstack und Hadoop, und entsprechender Programmierwerkzeuge, z.B. Pig und HBase, werden kennen gelernt und deren Einsatzgebiete diskutiert. Die Studierenden können also die Lösungen ganzheitlich analysieren und bewerten und somit technisch fundierte Entscheidungen für die Diensterbringung treffen.

Kompetenzen:

LE#	Lernergebnis	Geprüft durch
LE1	Verständnis der unterschiedlichen Cloud-Business-Modelle (IaaS, PaaS, SaaS) haben und anwenden können	Klausur
LE2	Komponenten und deren Aufgaben in einer Cloud-Architektur in Beziehung setzen können	Klausur
LE3	Operative Aspekte einer Cloud-Infrastruktur verstehen und bewerten können	Klausur
LE4	Methoden der Softwareentwicklung für eine Cloud-Plattform verstehen und anwenden können	Praktikum

Inhalt:

Die Dienstmodelle Infrastructure as a Service (IaaS), Platform as a Service (PaaS) sowie Software as a Service (SaaS) werden sowohl aus Anbietersicht als auch Nutzersicht dargestellt. Im Vordergrund stehen die Themen Softwareentwicklung für die Cloud als auch ein grundlegendes Verständnis für Cloud-Umgebungen. Die Liefermodelle Public Cloud, Private Cloud, Hybrid Cloud, Community Cloud werden anhand von Fallbeispielen vermittelt. Schwerpunktmäßig werden die Bezüge zu mobilen Anwendungen aufgezeigt. Ausführlich werden technische, organisatorische, kommerzielle, rechtliche, soziale und sicherheitsrelevante Aspekte des Cloud Computings behandelt, bewertet und diskutiert

Medienformen:

Seminaristische Vorlesung, Folien und Tafelanschrieb; Fallstudienbearbeitung in Kleingruppen.

Literatur:

- Antonopoulos, Nick; Gillam, Lee (2010): Cloud Computing. Principles Systems and Applications. London: Springer London (SpringerLink: Bücher, 0).

- Baun, Christian; Kunze, Marcel; Nimis, Jens; Tai, Stefan (2011): Cloud Computing. Web-basierte dynamische IT-Services. Berlin, Heidelberg: Springer Berlin Heidelberg (SpringerLink : Bücher).
- Buyya, Rajkumar (2011): Cloud computing. Principles and paradigms. Hoboken, NJ: Wiley (Wiley series on parallel and distributed computing).

Modul:	Mediale Arbeit	
Kürzel:	mkiB66	
Untertitel:		
Lehrveranstaltungen:	Vorlesung	
Studiensemester:	jedes Semester	
Modulverantwortliche:	Prof. Boris Terpinc	
Dozent(in):	Prof. Boris Terpinc Prof. Dr. Uwe Kloos Prof. Dr. Gabriela Tullius	
Sprache:	Deutsch, Englisch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 6. Semester	
Lehrform/SWS:	Vorlesung	2 SWS
Arbeitsaufwand:	Präsenzstudium	30 Stunden
	Eigenstudium	120 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung:	mkiB25, mkiB36, mkiB46	
Studien-/Prüfungsleistungen/ Prüfungsform:	Vorlesung: Continuous Assessment	

Modulziele:

Das Ziel des Moduls Mediales Arbeiten (MA) ist es, die Studierenden in die Realisierung von Beiträgen für die unterschiedlichsten Medienformen einzuführen. Im Modul mediale Arbeit kommen die bisher gelernten Inhalte aus den medialen Fächern wie Audio, Mensch-Maschine-Interaktion, Video und Computergrafik zusammen. Wurden bisher die Inhalte gemäß der inhaltlichen und thematischen Schwerpunkte in Projekten bearbeitet, wird hier nun übergreifend gearbeitet. Film/Video Berichterstattung, Journalistische Beiträge, Spielkonzepte, Kombinationen von Realfilm und Animationen werden dabei eingesetzt, um den Inhalt bestmöglich zu kommunizieren. Dabei kann es sich bei dem Inhalt auch um eine Spieleentwicklung oder ein Modell zur Simulation z.B. unter Nutzung existierender Game Engines handeln. Je nach Medienform und Inhalt müssen passende Interaktionsdesigns gewählt werden, so dass die Informationsvermittlung auch in interaktiven Medien effizient ist.

Der Fokus liegt dabei auf die inhaltliche Konzeption und Darstellung zum Zwecke der Informationsvermittlung. Das Modul ist bewusst breiter gehalten, um den Studierenden die Möglichkeit zu geben, eigene Schwerpunkte zu wählen.

Angestrebte Lernergebnisse:

Kenntnisse:

- Die Studenten arbeiten sich in Sach- und Fachthemen ein.
- Sie können optisch und inhaltlich recherchieren.
- Sie entwickeln passend zur Themenstellung Drehkonzept und Treatment und/oder Interaktions- und/oder Spielekonzept.
- Sie können themenbezogene Inhalte klar und verständlich vermitteln.
- Sie können abstrakter Inhalte in Bilder, Grafiken oder Animationen umsetzen.
- Sie können ein (Multimedia)Drehbuch erstellen und das auch weitestgehend mit Teams umsetzen.

Fertigkeiten:

Die Studierenden analysieren das Thema, die Nutzergruppe, den Kontext und die gestellte Aufgabe. Dazu wenden sie eine Reihe von erlernten Methoden an. Die Studierenden beurteilen die medialen Produkte nach wissenschaftlichen Kriterien.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage:

LE#	Lernergebnis (LE)	Geprüft durch
LE1	Recherche bzgl. Themas.	Hausarbeit
LE2	Zusammenarbeit und Arbeitsprozess der unterschiedlichen medialen Elemente, wie Video, Computergrafik oder Audio verstehen.	Referat, Projektarbeit
LE3	Entwicklung einer Konzeption für mediale Produkte.	Referat, Hausarbeit
LE4	Lauffähiges je nach Medienform prototypisches Projekt entwickeln.	Projektarbeit
LE5	Die entwickelten Projekte nach vorgegebenen Kriterien beurteilen.	Schriftliche Ausarbeitung

Inhalt:

Die Studierenden lernen sich mit den unterschiedlichen medialen Formen und deren jeweiligen Spezifika auseinanderzusetzen. Wie auch immer die mediale Form konkret aussieht – ob Dokumentarfilm, Spiel, Website oder mobile Anwendung, gemeinsam ist die Recherche zu Beginn, um das Thema genau zu recherchieren (LE1). Sie lernen Arbeitsprozesse zur Erstellung unterschiedlicher medialer Formen kennen und können dies auch anwenden (LE2, LE3). Die Entwicklung einer Konzeption ist für alle medialen Formen wichtig. Die Studierenden werden hier mit Möglichkeiten der Abstraktion der recherchierten Inhalte in unterschiedliche mediale Formen vertraut gemacht, um so den Inhalt bestmöglich zu kommunizieren. Die Umsetzung der medialen Arbeiten als Projekte erfolgt je nach Schwerpunkt (typischerweise Journalismus, Film, Simulation oder Spieleentwicklung) und wird durch die Projektarbeit geprüft (LE4). Durch die kritische Beurteilung und Auseinandersetzung mit den unterschiedlichen Projektarbeiten, überblicken die Studierenden die Vielfalt medialer Arbeiten mit den jeweiligen Besonderheiten (LE5).

Medienformen:

Seminaristischer Unterrichtsstil mit Beispielen. Projektarbeit im Team mit Betreuung durch die Dozenten. Aktivitäten zur Recherche, Konzeption, Dokumentation und Präsentation der Ergebnisse. Fachliche Arbeitsunterlagen sowie benötigte Hard- und Software werden bereitgestellt. Die erforderlichen Projektunterlagen sind in Eigeninitiative zu beschaffen und selbständig zu bearbeiten.

Literatur:

Wird in der Veranstaltung passend zum thematischen Schwerpunkt angegeben und durch die Studierenden bearbeitet.

Modul:	Psychologie	
Kürzel:	mkiB72	
Untertitel:		
Lehrveranstaltungen:	Vorlesung	
Studiensemester:	jedes Semester	
Modulverantwortliche:	Prof. Dr. Gabriela Tullius	
Dozent(in):	Dr. Andreas Rupp	
Sprache:	Deutsch, Englisch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 7. Semester	
Lehrform/SWS:	Vorlesung	4 SWS
Arbeitsaufwand:	Präsenzstudium	60 Stunden
	Eigenstudium	90 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung:	keine	
Studien-/Prüfungsleistungen Prüfungsform:	Vorlesung: Hausarbeit, Referat	

Modulziele:

Das Ziel des Moduls ist es, die Studierenden vertiefend in das Themenfeld der Psychologie, insbesondere der Medien- und Organisationspsychologie einzuarbeiten. Studierende lernen Methoden der Psychologie kennen und können nach erfolgreichem Bestehen des Moduls Verhaltensweisen von Menschen in Zusammenhang mit der Mediennutzung sowie des Verhaltens in Organisationsformen einschätzen.

Angestrebte Lernergebnisse:

Kenntnisse:

- Methoden aus dem Bereich der angewandten Medienpsychologie kennen lernen.
- Methoden aus der Organisationspsychologie kennen lernen.
- Aspekte der Mediennutzung bzgl. Wahrnehmung, Erleben und Verarbeitens von Medieninhalten einschätzen können.
- Aspekte bzgl. des Lernens in Organisationen, der Arbeitsanalyse und Personalentwicklung sowie Führungskulturen kennen und einschätzen lernen.

Fertigkeiten:

Die Studierenden werden mit den Methoden der Medien- und Organisationspsychologie vertraut gemacht. Studierenden erhalten einen Einblick in die beiden Themengebiete und lernen die Verbindungen, bspw. aus dem Bereich der Medienpädagogik oder der Auswertung kennen. Dazu gehören zum Beispiel Aspekte der menschlichen Verhaltensweisen, das menschliche Denken und Fühlen. Die Studierenden lernen Möglichkeiten der Weitervermittlung von Medienkompetenz auf individueller und organisatorischer Basis kennen und anzuwenden.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage:

LE#	Lernergebnis (LE)	Geprüft durch
LE1	Entwicklungen und Fähigkeiten von Individuen z.B. bzgl. medialer Inhalte beurteilen können.	Referat
LE2	Ansätze der Arbeitsanalyse und Personalentwicklung kennen.	Referat
LE3	Gängige Begriffe der Medienpsychologie und Organisationspsychologie erklären können (bspw. Massenmedien, Nutzungsformen von Medien, Coaching, Führungsformen).	Referat
LE4	Medienkompetenz erklären können.	Referat
LE5	Medienkompetenz vermitteln können.	Referat, Hausarbeit
LE6	Mediennutzungs- und auswahlverhalten kennen lernen.	Referat, Hausarbeit
LE7	Lerntheorien und ihre Anwendungsmöglichkeiten kennenlernen.	Referat, Hausarbeit
LE8	Konzepte der Personalentwicklung und Führung beschreiben können.	Referat, Hausarbeit

Inhalt:

Das Modul Psychologie hat die Schwerpunkte Medien- und Organisationspsychologie. Dies lässt sich dadurch begründen, dass diese beiden Teilgebiete der Psychologie neben den gemeinsamen Grundlagen bzgl. der Fähigkeiten von Individuen (LE1, LE2) einige gemeinsame Entwicklungsfelder haben (LE3). Hier ist beispielsweise das Gebiet der Lerntheorien und dessen Anwendung hinsichtlich medienpsychologischer Bereiche zu nennen (LE7).

Während im Bereich Medienpsychologie der Schwerpunkt bei der Medienkompetenz gesetzt wird (LE4-6), werden im Bereich Organisationspsychologie die Themen individuelles und organisationales Lernen im Unternehmen (LE2), Lerntheorien, Analyse und Gestaltung von Arbeitsumgebung und –aufgaben, Personalentwicklung, Führungskonzepte und Kommunikationsaspekte behandelt (LE2, LE8). Die Inhalte werden im Rahmen der Veranstaltung insbesondere durch Diskussionen und die Beiträge von Studierenden und Dozenten gefestigt.

Literatur:

- Gerrig, Richard J.; Zimbardo, Philip G.; Graf, Ralf (2008): Psychologie. 18., aktualisierte Aufl. München: Pearson Studium (ps psychologie).
- Mangold, Roland (2004): Lehrbuch der Medienpsychologie. Göttingen, Bern: Hogrefe Verl. für Psychologie.
- Rosenstiel, Lutz von; Nerdinger, Friedemann W. (2011): Grundlagen der Organisationspsychologie. Basiswissen und Anwendungshinweise. 7., überarb. Aufl. Stuttgart: Schäffer-Poeschel.
- Weitere Literatur wird in der Veranstaltung bekannt gegeben.

Modul:	Forschung und Entwicklung	
Kürzel:	mkiB73	
Untertitel:		
Lehrveranstaltungen:	Seminar	
Studiensemester:	jedes Semester	
Modulverantwortliche:	Prof. Dr. Benjamin Himpel	
Dozent(in):	Prof. Dr. Benjamin Himpel	
Sprache:	Deutsch, Englisch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 7. Semester	
Lehrform/SWS:	Vorlesung	2 SWS
Arbeitsaufwand:	Präsenzstudium	30 Stunden
	Eigenstudium	120 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung:	SAT	
Studien- und Prüfungsleistung:	Seminar: Referat, unbenotet	

Modulziele:

Das Ziel des Moduls ist es, die Studierenden an das Themenfeld wissenschaftliches und forschungsorientiertes Arbeiten begleitend zu Ihrer Bachelorthesis heranzuführen mit dem Ziel entwickelte oder zu entwickelnde Anwendungen im Kontext der Forschungsleistung zu sehen. Dabei steht Forschung und Entwicklung typischerweise für Ansätze aus der angewandten Forschung, wie sie oftmals in der Industrie stattfindet. Neben der angewandten Forschung wird auch die eher grundlagenorientierte Forschung betrachtet und Unterschiede aufgezeigt.

Angestrebte Lernergebnisse:

Kenntnisse:

- Formen und Methoden aus dem Bereich der angewandten Forschung kennen lernen.
- Lizenzmodelle kennen lernen.
- Den Wissenschaftsbetrieb und Möglichkeiten der Veröffentlichung von Arbeitsergebnissen kennen lernen.
- Formatvorlagen anwenden können.
- Rückmeldungen im Sinne eines Gutachter Reviews zu Arbeiten geben können.

- Wissenschaftliche Artikel und Studien aus dem eigenen Entwicklungsbereich einordnen können.
- Eigene Anwendungen für weitere Entwicklungen bewerten können.
- Informationen zum Thema Patente und Gebrauchsmuster recherchieren können.

Fertigkeiten:

Die Studierenden werden mit den Methoden bspw. zur Veröffentlichung von Konferenzbeiträgen vertraut gemacht. Von der Erstellung eines Beitrags bis hin zu Einreichung über elektronische Konferenzsysteme. Die Studierenden reichen bei einer fiktiven Konferenz selbst einen Konferenzbeitrag ein und lassen diesen von Ihren Kommilitonen begutachten und bewerten. Darüber hinaus werden die Studierenden gefördert, einen Beitrag bei einer realen Konferenz oder anderen wissenschaftlichen Publikationsformen einzureichen.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage:

LE#	Lernergebnis (LE)	Geprüft durch
LE1	Eigene Entwicklungen und Fähigkeiten im Vergleich zu anderen beurteilen können.	Referat, Diskussion
LE2	Konstruktive Rückmeldung zu Entwicklungen geben zu können.	Diskussion
LE3	Einen Beitrag so verfassen, dass die Annahme bei einer Konferenz möglich ist.	Referat, Diskussion
LE4	Erweiterung der Informationskompetenz insbesondere in Hinblick auf Information Retrieval Methoden zum Beispiel für Patente und Gebrauchsmuster.	Referat
LE5	Lizenzmodelle für unterschiedliche Artefakte beurteilen und anwenden zu können.	Referat, Diskussion

Inhalt:

Die Veranstaltung hat seminaristischen Charakter. Die Studierenden sind aufgefordert sich durch Referate und andere Wort- sowie Schriftbeiträge zu beteiligen (LE1). Ausgehend von einer eigenen Forschungsarbeit, die typischerweise im Rahmen der Bachelor Thesis erbracht wird, lernen die Studierenden Möglichkeiten der Veröffentlichung von Arbeitsergebnissen kennen (LE 3). Durch den Aufbau und die Nutzung eines Konferenzsystems zur Begutachtung von Beiträgen lernen die Studierenden Möglichkeiten des „Peer Review“ kennen (LE2). Dabei nutzen die Studierenden ein Konferenzsystem für eine fiktive Konferenz als Übungsszenario, um einerseits selbst Beiträge zu begutachten und andererseits eigene Beiträge begutachten zu lassen. Das Thema Information Retrieval, insbesondere von Patentinformation rundet die Veranstaltung ab (LE4). Die Auseinandersetzung mit Themen wie Open Access und Lizenzmodelle wie Creative Commons u.a. vervollständigt diesen Teil der Informationskompetenz (LE5). In einem Referat zu dem methodischen Vorgehen und die Zusammenfassung der Ergebnisse berichten die Studierenden über Ihre eingesetzten Methoden, Erfahrungen und Möglichkeiten der Weiterverwendung ihrer Ergebnisse (LE1).

Literatur:

- Esselborn-Krumbiegel, Helga (2008): Von der Idee zum Text. Eine Anleitung zum wissenschaftlichen Schreiben. 3., überarb. Aufl. Paderborn, München, Wien, Zürich: Schöningh (utb.de Bachelor-Bibliothek, 2334 : Schlüsselkompetenzen, Kernkompetenzen).
- Kuhlen, Rainer; Semar, Wolfgang; Strauch, Dietmar (Hg.) (2014): Grundlagen der praktischen Information und Dokumentation. Handbuch zur Einführung in die Informationswissenschaft und -praxis. 6., völlig neu gefasste Ausg. Berlin: De Gruyter.

Weitere Literatur zu den Themen Patentinformation, Creative Commons, Softwarelizenzen werden in der Veranstaltung bekannt gegeben.

Modul:	Bachelor Kolloquium	
Kürzel:	mkiB74	
Untertitel:		
Lehrveranstaltungen:	Kolloquium	
Studiensemester:	jedes Semester	
Modulverantwortliche:	Prof. Dr. Natividad Martínez	
Dozent(in):	Professorinnen und Professoren des Studienganges	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 7. Semester	
Lehrform/SWS:	Vorlesung	2 SWS
Arbeitsaufwand:	Präsenzstudium	30 Stunden
	Eigenstudium	60 Stunden
Kreditpunkte:	3 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung:	Veranstaltungen der ersten 5 Fachsemester	
Studien-/Prüfungsleistungen/ Prüfungsform:	Kolloquium: Referat, Teilnahme im 6. und 7. Semester	

Modulziele:

Ziel ist ein Fachgespräch über die Themen der Bachelorarbeiten. Jeder Student, der eine Bachelor-Thesis durchführt, hält einmal einen Vortrag über den aktuellen Stand seiner Thesis, um andere Studierende und Lehrende zu informieren, seine Arbeit kritisch zu würdigen und Feedback zu erhalten.

Angestrebte Lernergebnisse:

Kenntnisse:

- Technisch-wissenschaftlichen Vortrag halten (keine Werbeveranstaltung).
- Prägnante Einführung in das Thema/Motivation.
- Beschreibung der Methoden und Vorgehensweisen/Planung.
- Strukturierte Ausarbeitung der Kernpunkte/Niveau der Argumentation/Korrektheit vorgebrachter Behauptungen.
- Überzeugende Darstellung der mit der Bachelorarbeit selbst geleisteten Arbeit.

- Überzeugender verbaler und persönlicher Vortragsstil.
- Verständlichkeit des Vortrags für Fachleute, die das Thema nicht kennen.
- Angemessenheit der Gestaltung und des Medieneinsatzes (Folien, Online-Präsentation)/Qualität der präsentierten/kopierten Folien und Handzettel.
- Qualität der Literaturangaben.
- Vorbereitung und Führung der Diskussion.
- Einhaltung und effiziente Nutzung der Vortragszeit.
- Aktive Teilnahme am Fachgespräch.

Fertigkeiten:

Die Studierenden entwickeln ein Konzept zur geeigneten Darstellung ihres Themas. Die zuhörenden Studierenden beurteilen die Angemessenheit der Gestaltung und des Vortragsstils. Studierende – sowohl Vor- als auch Zuhörende erfahren die Bedeutung von Fachgesprächen und Argumentationslinien. Der Student übt, Ergebnisse eigener Arbeit Anderen verständlich, strukturiert und prägnant darzustellen.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage:

LE#	Lernergebnis (LE)	Geprüft durch
LE1	Soziale und kommunikative Kompetenz: Eigene Arbeiten Anderen verständlich, strukturiert und prägnant darzustellen.	Referat
LE2	Genaue Beschreibung der Aufgaben, Anforderungen, Ziele, Methoden, Randbedingungen, klare Abgrenzung der Aufgaben und durchgeführten Arbeiten vom vorgefundenen Umfeld.	Referat
LE3	Vortragsplanung Planung und Durchführung. Strukturierte Ausarbeitung der Kernpunkte.	Referat
LE4	Diskussionsführung und Argumentation.	Diskussion aller Anwesenden nach jeweiligem Vortrag

Inhalt:

Das Bachelor-Kolloquium ist inhaltlich mit den Themen aller Bachelorarbeiten verbunden. Jeder Student, der eine Bachelor-Thesis durchführt, präsentiert mit einem Vortrag den aktuellen Stand seiner Thesis (LE2 und LE3), um andere Studierende und Lehrende zu informieren (LE1), Feedback zu erhalten und Anregungen aufzunehmen (LE4). Je nach Kenntnisstand kann er das Thema der Bachelorarbeit vorstellen, in das (betriebliche) Umfeld und den Stand der Wissenschaft einordnen, eine Literaturübersicht geben, gesteckte Ziele, gestellte Aufgaben, anzuwendende Konzepte, Methoden, Vorgehensweisen beschreiben, über den erreichten Stand berichten, gelöste/ungelöste Aufgaben diskutieren, oder die ganze Arbeit mit Aufgabenstellung, Lösungsansätzen, Ergebnissen, Fazit und Ausblick zusammenfassen.

Medienformen:

Betreuung der Vorbereitung des Vortrags durch den Betreuer der Bachelor-Thesis. Durchführung Vortrag mit Diskussionsforum. Rückmeldung von allen anwesenden Studierenden.

Literatur:

- Deininger, Marcus (1993): Studien-Arbeiten. Ein Leitfaden zur Vorbereitung Durchführung und Betreuung von Studien- Diplom- und Doktorarbeiten am Beispiel Informatik. 2., durchges. Aufl. Zürich, Stuttgart: vdf; Teubner.
- Esselborn-Krumbiegel, Helga (2008): Von der Idee zum Text. Eine Anleitung zum wissenschaftlichen Schreiben. 3., überarb. Aufl. Paderborn, München, Wien, Zürich: Schöningh (utb.de Bachelor-Bibliothek, 2334 : Schlüsselkompetenzen, Kernkompetenzen).
- Leopold-Wildburger, Ulrike; Schütze, Jörg (2002): Verfassen und Vortragen. Wissenschaftliche Arbeiten und Vorträge leicht gemacht. Berlin, Heidelberg: Springer (Springer-Lehrbuch).

Modul:	Bachelor Thesis
Kürzel:	mkiB75
Untertitel:	
Lehrveranstaltungen:	Thesis
Studiensemester:	jedes Semester
Modulverantwortlicher:	Prof. Dr. Peter Hertkorn
Dozent(in):	Professorinnen und Professoren des Studienganges
Sprache:	Deutsch
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Pflichtfach, 7. Semester
Lehrform/SWS:	Thesis
Arbeitsaufwand:	Eigenstudium 360 Stunden
Kreditpunkte:	12 ECTS
Voraussetzungen nach StuPro:	alle Module der ersten 5 Semester sind bestanden
Empfohlene Voraussetzung	alle Module der ersten 6 Semester
Studien-/Prüfungsleistungen/ Prüfungsform:	Bachelor Thesis

Modulziele:

Durch erfolgreiches Bestehen des Moduls zeigt der Studierende, dass er ein Problem der Medien- und Kommunikationsinformatik selbstständig nach grundlegenden wissenschaftlichen Methoden fristgerecht bearbeiten kann.

Die Bachelor-Thesis trägt zu den Gesamtlehrzielen von Medien- und Kommunikationsinformatik wie folgt bei:

- Breites interdisziplinäres Fachwissen und umfassende Methodenkompetenz: Bachelorarbeiten erfordern, Kenntnisse und Methoden aus verschiedenen Disziplinen anzuwenden. Sie umfassen informatische, softwaretechnische, mediale, psychologische, didaktische, wirtschaftliche und andere Aspekte.
- Attraktive Berufsperspektive: Bachelorarbeiten befassen sich oft mit Problemen, die in der betrieblichen Praxis der Informatik aktuell relevant sind. Bachelorarbeiten können als externe Arbeiten in Kooperation mit Firmen durchgeführt werden.
- Internationalität: Bachelorarbeiten können in englischer Sprache verfasst werden. Sie können auch in Kooperation mit ausländischen Institutionen durchgeführt werden.

Angestrebte Lernergebnisse:

Kenntnisse:

- Recherchieren nach seriösen Quellen.
- Korrektes Zitieren von Textabschnitten.
- Referenzieren von Quellen.
- Präzises Darstellen eines Themas, des Kontexts und des Stands der Wissenschaft.
- Klares Formulieren einer Forschungsfrage und der Ziele einer Arbeit.
- Genaues Beschreiben von Methoden und Vorgehensweisen, sowie der Entwicklung von Artefakten.
- Strukturiertes Ausarbeiten von Kernpunkten.
- Schlüssiges Argumentieren und Begründen von Behauptungen.
- Überzeugendes und verständliches Darstellen der geleisteten Arbeit.

Fertigkeiten:

Die Studierenden führen eine Literaturrecherche nach wissenschaftlichen Quellen durch. Sie bereiten den Stand des Wissens kritisch auf. Sie analysieren Probleme, stellen Hypothesen auf, definieren Anforderungen und leiten Kriterien ab, nach denen Alternativen systematisch evaluiert werden. Die Studierenden strukturieren Problemstellungen in Teilaufgaben, entwickeln Lösungskonzepte und überprüfen kritisch die Ergebnisse. Sie realisieren Prototypen oder einsatzfähige Artefakte. Die Studierenden kommunizieren die Ergebnisse klar und in akademisch angemessener Form.

Kompetenzen:

LE#	Lernergebnis (LE)	Geprüft durch
LE1	Selbständige wissenschaftliche Bearbeitung eines Themas der Medien- und Kommunikationsinformatik	Thesis-Dokument
LE2	Arbeiten nach grundlegenden Methoden an einem einfachen Problem und kleinen Artefakt	Thesis-Dokument
LE3	Eigenständiges Verfassen einer wissenschaftliche Arbeit	Thesis-Dokument
LE4	Eigene Arbeiten verständlich, strukturiert und prägnant darstellen	Thesis-Dokument

Inhalt:

Bachelorarbeiten behandeln meist praktische, teilweise theoretische Probleme und Lösungsansätze aus der Medien- und Kommunikationsinformatik.

Medienformen:

Fachliche und methodische Betreuung der Bachelorarbeit durch Gespräche und Kommentare zu Entwürfen.

Literatur:

- Deininger, Marcus (2005): Studien-Arbeiten. Ein Leitfaden zur Vorbereitung Durchführung und Betreuung von Studien- Diplom- Abschluss- und Doktorarbeiten am Beispiel Informatik. 5., überarb. Aufl. Zürich: vdf Hochschulverl. an der ETH.
- Ebel, Hans F.; Bliefert, Claus (2009): Bachelor-, Master- und Doktorarbeit. Anleitungen für den naturwissenschaftlich-technischen Nachwuchs. 4., aktualisierte Auflage. Weinheim: Wiley-VCH.
- Esselborn-Krumbiegel, Helga (2008): Von der Idee zum Text. Eine Anleitung zum wissenschaftlichen Schreiben. 3., überarb. Aufl. Paderborn, München, Wien, Zürich: Schöningh (utb.de Bachelor-Bibliothek, 2334 : Schlüsselkompetenzen, Kernkompetenzen).
- Grieb, Wolfgang; Slemeyer, Andreas (2012): Schreibtipps für Studium, Promotion und Beruf in Ingenieur- und Naturwissenschaften. 7. Aufl. Berlin: VDE-Verl.
- Karmasin, Matthias; Ribing, Rainer (2012): Die Gestaltung wissenschaftlicher Arbeiten. Ein Leitfaden für Seminararbeiten Bachelor- Master- und Magisterarbeiten sowie Dissertationen. 7., aktualisierte Aufl. Wien: facultas.wuv (UTB, 2774 : Schlüsselkompetenzen).

Modul:	Medien- und Kommunikationsinformatik Projekt 1	
Kürzel:	mkiBW101	
Untertitel:		
Lehrveranstaltungen:	Praktikum	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Dr. Peter Hertkorn	
Dozent(in):	Professorinnen und Professoren des Studienganges	
Sprache:	Deutsch, Englisch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Wahlpflichtfach, 6. Semester	
Lehrform/SWS:	Praktikum	2 SWS
Arbeitsaufwand:	Präsenzstudium	30 Stunden
	Eigenstudium	120 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	Keine	
Empfohlene Voraussetzung :	Keine	
Studien-/Prüfungsleistungen/ Prüfungsform:	Praktikum: Praktikum	

Modulziele

Das Ziel des Moduls ist es, dass die Studierenden die im Studium erworbenen Kenntnisse und Fertigkeiten in einem praktischen Fallbeispiel auf Problemstellungen aus dem Bereich der Medien- und Kommunikationsinformatik anwenden. Diese umfassen typischerweise sowohl mediale Themenbereiche als auch softwaretechnische Umsetzungen. Insbesondere sind hierbei auch die Kommunikations- und Teamfähigkeit der Studierenden gefordert, da die Studierenden die Problemstellungen selbständig in Arbeitsgruppen bearbeiten.

Im ersten Teil des Projekts liegt der Schwerpunkt auf der Erstellung von Prototypen und dem dazugehörigen Entwicklungsprozess.

Angestrebte Lernergebnisse

Kenntnisse:

Vertiefung der im Studium erworbenen Kenntnisse zu den einzelnen Schritten im Entwicklungsprozess:

- Vorgehens- und Prozessmodelle.
- Methoden zur Projektplanung und –verwaltung.
- Techniken der Anforderungsanalyse.
- Verfahren zur Aufwandsschätzung.
- Methoden und Verfahren zur Modellierung, dem Entwurf und der Realisierung von Systemen.
- Einsatz unterschiedlicher Testverfahren.
- Methoden zur Dokumentation der Ergebnisse.

Fertigkeiten:

Die Studierenden analysieren eine gegebene Problemstellung und wenden Methoden und Verfahren für die Definition der Anforderungen, die Erstellung von Modellen, dem Entwurf und der Realisierung von Systemen sowie für die Durchführung von Tests an. Sie analysieren Probleme, bewerten verschiedene Lösungsalternativen und entwickeln eigenständig Lösungen. Die Studierenden arbeiten selbständig innerhalb eines Projektteams, wobei sie das Projekt planen, die Arbeit aufteilen, sowie die Ergebnisse zu einem späteren Zeitpunkt wieder zusammenführen. Sie können die Ergebnisse der Teamarbeit angemessen dokumentieren und einem Fachpublikum verständlich präsentieren.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage:

LE#	Lernergebnis (LE)	Geprüft durch
LE1	Methoden und Verfahren für die systematische Entwicklung von Produkten bzw. Softwaresystemen anzuwenden.	Artefakt
LE2	Eine Problemstellung zu analysieren, Lösungsalternativen zu erarbeiten und zu bewerten, daraus eigene Verfahren zu entwickeln und in Diskussionen fundiert zu argumentieren.	Artefakt
LE3	Die entwickelten Lösungen zu dokumentieren und einem Fachpublikum zu präsentieren.	Artefakt, Referat
LE4	Im Team zu arbeiten, gemeinsam Ziele zu definieren, diese zu verfolgen und einzuhalten.	Artefakt, Referat
LE5	Professionelle Werkzeuge für den gesamten Entwicklungsprozess einsetzen zu können.	Artefakt

Inhalt:

Aus einer Reihe von Projektthemen, die von den Dozenten angeboten werden, wählen die Projektteams ein Arbeitsthema aus. In Abstimmung mit den Betreuern strukturieren die Studierenden das Projekt in geeignete Entwicklungsphasen. Dabei wird ein Arbeits- und Zeitplan für das Projekt festgelegt in dem auch die regelmäßigen Besprechungen mit den Betreuern geplant werden. Zu den jeweiligen Meilensteinen werden die erzielten Ergebnisse dokumentiert und präsentiert sowie der Stand der Prototypen vorgestellt.

Medienformen:

Projektarbeit im Team mit Betreuung des Dozenten. Aktivitäten zur Spezifikation, Entwicklung, Test, Dokumentation und Präsentation der erarbeiteten Ergebnisse. Fachliche Arbeitsunterlagen sowie benötigte Hard- und Software werden bereitgestellt. Die erforderlichen Projektunterlagen sind in Eigeninitiative zu beschaffen und selbständig zu bearbeiten.

Literatur

Die Literatur wird während der Projekte bekannt gegeben und durch die Studierenden selbstständig recherchiert.

Modul:	Medien- und Kommunikationsinformatik Projekt 2	
Kürzel:	mkiBW201	
Untertitel:		
Lehrveranstaltungen:	Praktikum	
Studiensemester:	jedes Semester	
Modulverantwortlicher:	Prof. Dr. Peter Hertkorn	
Dozent(in):	Professorinnen und Professoren des Studienganges	
Sprache:	Deutsch, Englisch	
Zuordnung zum Curriculum:	Medien- und Kommunikationsinformatik Bachelor, Wahlpflichtfach, 7. Semester	
Lehrform/SWS:	Praktikum	2 SWS
Arbeitsaufwand:	Präsenzstudium	30 Stunden
	Eigenstudium	120 Stunden
Kreditpunkte:	5 ECTS	
Voraussetzungen nach StuPro:	keine	
Empfohlene Voraussetzung :	Medien- und Kommunikationsinformatik Projekt 1	
Studien-/Prüfungsleistungen/ Prüfungsform:	Praktikum: Praktikum	

Modulziele

Das Ziel des Moduls ist es, dass die Studierenden die im Studium erworbenen Kenntnisse und Fertigkeiten in einem praktischen Fallbeispiel auf Problemstellungen aus dem Bereich der Medien- und Kommunikationsinformatik anwenden. Diese umfassen typischerweise sowohl mediale Themenbereiche als auch softwaretechnische Umsetzungen. Insbesondere sind hierbei auch die Kommunikations- und Teamfähigkeit der Studierenden gefordert, da die Studierenden die Problemstellungen selbständig in Arbeitsgruppen bearbeiten.

Im zweiten Teil des Projekts liegt der Schwerpunkt auf der Weiterentwicklung von Prototypen zu einem Produkt sowie der Einführung des Produkts in eine Betriebsumgebung.

Angestrebte Lernergebnisse

Kenntnisse:

- Methoden zur Weiterentwicklung von Prototypen zu einem Produkt kennen.
- Verfahren für die Bewertung der Benutzerfreundlichkeit kennen.

- Aspekte, die bei der Einführung von Produkten bzw. Installation von Softwaresystemen in Betriebsumgebungen beachtet werden müssen, kennen.
- Methoden zur Präsentation und dem Marketing von Produkten kennen.
- Informationen zum Thema Lizenzen und Patente recherchieren können.

Fertigkeiten:

Die Studierenden wenden Methoden und Verfahren für die Weiterentwicklung von Prototypen zur Produktreife an. Sie führen Tests zur Bewertung der Benutzerfreundlichkeit durch und leiten daraus Änderungen für die Produktentwicklung ab. Die Studierenden führen Produkte in bestehende Betriebsumgebungen ein und bauen geeignete Support-Strukturen auf. Sie erstellen Präsentationen für das Produkt und recherchieren Lizenzbestimmungen sowie gewerbliche Schutzrechte. Die Studierenden arbeiten selbständig innerhalb eines Projektteams, können die Ergebnisse der Teamarbeit angemessen dokumentieren und einem Fachpublikum verständlich präsentieren.

Kompetenzen:

Nach Abschluss des Moduls sind die Studierenden in der Lage:

LE#	Lernergebnis (LE)	Geprüft durch
LE1	Methoden und Verfahren für die Entwicklung eines Prototyps zur Produktreife anzuwenden.	Artefakt
LE2	Tests für die Bewertung der Benutzerfreundlichkeit eines Produkts durchzuführen.	Artefakt
LE3	Ein Produkt in bestehende Betriebsumgebungen einzuführen.	Artefakt, Referat
LE4	Die erarbeiteten Ergebnisse zu dokumentieren und einem Fachpublikum zu präsentieren.	Artefakt, Referat
LE5	Im Team zu arbeiten, gemeinsam Ziele zu definieren, diese zu verfolgen und einzuhalten.	Artefakt, Referat
LE6	Professionelle Werkzeuge für Installation, Betrieb und Produktpräsentation einsetzen zu können.	Artefakt

Inhalt:

Die Projektteams entwickeln den Prototypen aus dem ersten Teil des Medien- und Kommunikationsinformatik Projekts zur Produktreife weiter. In Abstimmung mit den Betreuern strukturieren die Studierenden das Projekt in geeignete Phasen. Dabei wird ein Arbeits- und Zeitplan für das Projekt festgelegt in dem auch die regelmäßigen Besprechungen mit den Betreuern geplant werden. Zu den jeweiligen Meilensteinen werden die erzielten Ergebnisse dokumentiert und präsentiert sowie der Stand des Produkts vorgestellt.

Medienformen:

Projektarbeit im Team mit Betreuung des Dozenten. Aktivitäten zur Spezifikation, Entwicklung, Test, Dokumentation und Präsentation der erarbeiteten Ergebnisse. Fachliche Arbeitsunterlagen sowie benötigte Hard- und Software werden bereitgestellt. Die erforderlichen Projektunterlagen sind in Eigeninitiative zu beschaffen und selbständig zu bearbeiten.

Literatur

Die Literatur wird während der Projekte bekannt gegeben und durch die Studierenden selbstständig recherchiert.