

# **Modulhandbuch für den Bachelor-Studiengang Medieninformatik**

**Anlage zur Studien- und Prüfungsordnung  
Bachelor Medieninformatik**

**vom 14.06.2016  
(A.M. 37/2016)**

**Stand: 08.12.2023**

**Ansprechpartner\*innen:**

**Dekan\*in FB VI  
fb6@bht-berlin.de**

**Prof. Dr. Simone Strippgen (Studiengangsleitung)  
strippgen@bht-berlin.de**

## Inhalt

Koordinatorinnen und Koordinatoren .....	5
<b>Pflichtmodule</b>	
Mathematik I / Mathematics 1 .....	7
Grundlagen der Theoretischen Informatik / Principles of Theoretical Computer Science.....	8
Mediendesign Grundlagen / Principles of Media Design.....	9
Technische Grundlagen der Informatik / .....	10
Technical Principles of Computer Science .....	10
Programmierung I / Programming 1 .....	11
Mathematik II / Mathematics 2 .....	12
Algorithmen und Datenstrukturen / Algorithms and Data Structures .....	13
Datenbanksysteme / Database Systems .....	14
Programmierung II / Programming 2 .....	15
Betriebssysteme / Operating Systems .....	16
Studium Generale I / General Studies 1.....	17
Studium Generale II / General Studies 2.....	18
Software Engineering I / Software Engineering 1 .....	19
Computergrafik Grundlagen / Principles of Computer Graphics.....	20
Medientechnologien / Media Technologies .....	21
Verteilte Systeme / Distributed Systems .....	22
Web Engineering I / Web Engineering 1 .....	23
Software Engineering II / Software Engineering 2 .....	24
Web Engineering II / Web Engineering 2 .....	25
Human-Computer Interaction / Human-Computer Interaction.....	26
Wahlpflichtmodul I / Required-Elective Module 1 .....	27
Wahlpflichtmodul II / Required-Elective Module 2 .....	28
Projekt / Project.....	29
Grundlagen wissenschaftlichen Arbeitens / Principles of Scientific Work.....	30
Wahlpflichtmodul III / Required-Elective Module 3 .....	31
Wahlpflichtmodul IV / Required-Elective Module 4.....	32
Praxisprojekt / Practice-Based Project .....	33
Abschlussprüfung / Final Examination Module.....	34

**Wahlpflichtmodule**

Interaktions- und Interface Design / Interaction and Interface Design .....	35
Software Engineering: Architekturen und Werkzeuge / Software Engineering: Architectures and Tools .....	37
3D Web Graphics / 3D Web Graphics.....	38
Softwarequalität und -test / Software Quality and Test .....	39
Betriebswirtschaftslehre / Business Administration .....	40
Medienproduktion und -distribution / Media Production and Distribution.....	41
Effiziente Software Entwickeln mit C++ / Developing Efficient Software with C++ .....	42
Programmiersprachen und -paradigmen / Programming Languages and Paradigms.....	43
Anwendungsentwicklung für IOS-Geräte / Application Development for IOS Devices .....	44
Mobile Anwendungsentwicklung / Mobile Application Development .....	45
Visual and Scientific Computing / Visual and Scientific Computing.....	47
Spielentwicklung und Creative Coding / Game Development and Creative Coding.....	48
Signalverarbeitung für Audio, Bild und Video / Signal Processing for Audio, Image and Video .....	49
Maschinelles Lernen / Machine Learning .....	50
Aktuelle Webtechnologien: Frameworks und Tools / Current Web Technologies: Frameworks and Tools .....	51
Aktuelle Themen / Selection of Current Issues .....	52
Frontend-Design Web: Fortgeschrittene Techniken /.....	53
Frontend-Design Web: Advanced Techniques.....	53
Microservice-Entwicklung: Architekturen und Entwurfsmuster / Developing Microservices: Architectures and Design Patterns .....	54
Interactive Media Objects & Web-Media Applications.....	55
Webprogrammierung mit Python / Web Programming with Python .....	56
Echtzeitgrafik Grundlagen / Fundamentals of Real-Time Rendering .....	57
Professionelle Web-Anwendungen umsetzen und betreiben / Implementing and maintaining professional web applications.....	58
Data Science Programmierung mit Python / Data Science Programming with Python .....	59

## Koordinatorinnen und Koordinatoren

### Pflichtmodule

Modul	Modulname	Koordinator/in
B01	Mathematik I	Prof. Dr. Martin Oellrich (FB II), Studiengangsleitung
B02	Grundlagen der Theoretischen Informatik	Prof. Dr. Heike Ripphausen-Lipa
B03	Mediendesign Grundlagen	Prof. Pamela Schaudin
B04	Technische Grundlagen der Informatik	Prof. Dr. Peter Gregorius
B05	Programmierung I	Prof. Dr. Simone Strippgen
B06	Mathematik II	Prof. Dr. Martin Oellrich (FB II), Studiengangsleitung
B07	Algorithmen und Datenstrukturen	Prof. Dr. Heike Ripphausen-Lipa
B08	Datenbanksysteme	Prof. Dr. Petra Sauer
B09	Programmierung II	Prof. Dr. Simone Strippgen
B10	Betriebssysteme	Prof. Dr. Rüdiger Weis
B11	Studium Generale I	Dekan/Dekanin FB I
B12	Studium Generale II	Dekan/Dekanin FB I
B13	Software Engineering I	Prof. Dr. Dragan Macos
B14	Computergrafik Grundlagen	Prof. Dr. Henrik Tramberend
B15	Medientechnologien	Prof. Dr. Hansjörg Mixdorff
B16	Verteilte Systeme	Prof. Dr. René Görlich
B17	Web Engineering I	Prof. Dr. Simone Strippgen
B18	Software Engineering II	Prof. Dr. Dragan Macos
B19	Web Engineering II	Prof. Dr. Sebastian von Klinski
B20	Human-Computer Interaction	Prof. Dr. Robert Strzebowski
B21	Wahlpflichtmodul I	
B22	Wahlpflichtmodul II	
B23	Projekt	Prof. Dr. Hartmut Schirmacher
B24	Grundlagen wissenschaftlichen Arbeitens	Prof. Dr. Antje Ducki, Studiengangsleitung
B25	Wahlpflichtmodul III	
B26	Wahlpflichtmodul IV	
B27	Praxisprojekt	Studiengangsleitung
B28	Abschlussprüfung	Studiengangsleitung

**Wahlpflichtmodule**

<b>Modul</b>	<b>Modulname</b>	<b>Koordinator/in</b>
WP01	Interaktions- und Interfacedesign	Prof. Pamela Schaudin
WP02	Software Engineering: Architekturen und Werkzeuge	Prof. Dr. Dragan Macos
WP03	3D Web Graphics	Prof. Dr. Henrik Tramberend
WP04	Softwarequalität und -test	Prof. Dr. Sven Graupner
WP05	Betriebswirtschaftslehre	Prof. Dr. Alexander Huber (FB I)
WP06	Medienproduktion und -distribution	Prof. Dr. Jürgen Lohr
WP07	Effiziente Software Entwickeln mit C++	Prof. Dr. Hartmut Schirmacher
WP08	Programmiersprachen und -paradigmen	Prof. Dr. Dragan Macos
WP09	Anwendungsentwicklung für iOS-Geräte	Prof. Dr. Dragan Macos
WP10	Mobile Anwendungsentwicklung	Prof. Dr. Jörn Kreutel
WP11	Visual and Scientific Computing	Prof. Dr. Kristian Hildebrand
WP12	Spieleentwicklung und Creative Coding	Prof. Dr. Kristian Hildebrand
WP13	Signalverarbeitung für Audio, Bild und Video	Prof. Dr. Jürgen Lohr
WP14	Maschinelles Lernen	Prof. Dr. Felix Bießmann
WP15	Aktuelle Webtechnologien: Frameworks und Tools	Prof. Dr. Sebastian von Klinski
WP16	Aktuelle Themen	Studiengangsleitung
WP17	Frontend-Design Web: Fortgeschrittene Techniken	Studiengangsleitung
WP18	Microservice-Entwicklung: Architekturen und Entwurfsmuster	Studiengangsleitung
WP19	Interactive Media Objects & Web-Media Applications	Prof. Dr. Jürgen Lohr
WP20	Webprogrammierung mit Python	Prof. Dr. Amy Siu
WP21	Echtzeitgrafik Grundlagen	Prof. Dr. Henrik Tramberend
WP22	Professionelle Web-Anwendungen umsetzen und betreiben	Prof. Dr. Sebastian von Klinski
WP23	Data-Science Programmierung mit Python	Prof. Dr. Amy Siu

Gesamtansprechpartner für das Modulhandbuch:

Dekanin: Prof. Dr. Petra Sauer (Email: [sauer@bht-berlin.de](mailto:sauer@bht-berlin.de))

Studiengangsleitung: Prof. Dr. Simone Strippgen (Email: [strippgen@bht-berlin.de](mailto:strippgen@bht-berlin.de))

<b>Modulnummer</b>	<b>B01</b>
<b>Titel</b>	<b>Mathematik I / Mathematics 1</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	102 Stunden Präsenz (4 SWS SU + 2 SWS Ü), 48 Stunden Selbststudium
<b>Lerngebiet</b>	Mathematisch-naturwissenschaftliche Grundlagen
<b>Lernziele / Kompetenzen</b>	Kenntnisse in den mathematischen Grundlagen der Informatik, Sicherheit im abstrakten und strukturellen Denken.
<b>Voraussetzungen</b>	Empfehlung: die Belegung des Mathematik-Brückenkurses vor Studienbeginn
<b>Niveaustufe</b>	1. Studienplansemester
<b>Lehrform</b>	Seminaristischer Unterricht, Übung
<b>Status</b>	Pflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 100% Klausur (90 Minuten).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<p>I Diskrete Mathematik</p> <ol style="list-style-type: none"> <li>1. Aussagenlogik <ul style="list-style-type: none"> <li>• Aussagen und ihre Verknüpfungen, Syntax und Semantik von Ausdrücken, Tautologien, semantische Äquivalenz und Implikation, disjunktive und konjunktive Normalformen.</li> </ul> </li> <li>2. Prädikatenlogik <ul style="list-style-type: none"> <li>• Prädikate und Quantoren, freie und gebundene Variablen, Syntax und Semantik von prädikatenlogischen Ausdrücken, Tautologien</li> <li>• Beweisverfahren in der Mathematik</li> </ul> </li> <li>3. Grundbegriffe der Mengenlehre <ul style="list-style-type: none"> <li>• Cantorscher Mengenbegriff, Gleichheit, Teilmengen, Potenzmengen</li> <li>• Mengenoperationen und ihre Gesetze</li> <li>• Mächtigkeit, endliche, unendliche Mengen; IN, IZ, IQ und IR</li> </ul> </li> <li>4. Kombinatorik <ul style="list-style-type: none"> <li>• Zählprinzipien: Modulgleichung, Summen- und Produktregel</li> <li>• Bijektionen endlicher Mengen, Permutationen, Potenzmenge</li> <li>• Fakultät und Binomialkoeffizienten, Binomischer Satz</li> </ul> </li> <li>5. Relationen und Abbildungen <ul style="list-style-type: none"> <li>• Darstellung von Relationen auf endlichen Mengen durch Matrizen und Graphen, Inversion und Verkettung</li> <li>• Eigenschaften von Relationen auf einer Menge: reflexiv, irreflexiv, symmetrisch, antisymmetrisch, transitiv, linear; Hüllenbildung</li> <li>• Äquivalenz- und Ordnungsrelationen, Hasse-Diagramme</li> <li>• Funktionen: Eigenschaften (linkstotal/rechtseindeutig, injektiv, surjektiv, bijektiv)</li> </ul> </li> </ol>
<b>Literatur</b>	<p>G. + S. Teschl: <i>Mathematik für Informatiker Band 1</i>, Springer+Vieweg Verlag  P. Hartmann: <i>Mathematik für Informatiker</i>, Vieweg+Teubner Verlag  L. Görke: <i>Mengen, Relationen, Funktionen</i>, Harri Deutsch Verlag</p>
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>B02</b>
<b>Titel</b>	<b>Grundlagen der Theoretischen Informatik / Principles of Theoretical Computer Science</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (2 SWS SU + 2 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Ziel ist das Erlernen der Grundlagen von Automaten und formalen Sprachen als Werkzeug zur Modellierung und Transformation von Systemen und Prozessen. Diese Grundlagen sollen ermöglichen den Aufbau von Programmiersprachen besser zu verstehen und die Grenzen von Programmen beurteilen zu können.
<b>Voraussetzungen</b>	keine
<b>Niveaustufe</b>	1. Studienplansemester
<b>Lehrform</b>	Seminaristischer Unterricht, Übung
<b>Status</b>	Pflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 100% Klausur (90 Minuten).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<ol style="list-style-type: none"> <li>1. Alphabete und formale Sprachen Grundbegriffe, Zusammenhang Programmiersprachen</li> <li>2. Endliche Automaten Syntax, Semantik und Beispiele, Anwendungen von endlichen Automaten (z.B. UML-Modellierung, Zustandsübergangdiagramm, Beschreibung des Verhaltens von Hardware), deterministische endliche Automaten, nichtdeterministische endliche Automaten, minimale endliche Automaten</li> <li>3. Reguläre Ausdrücke Syntax, Semantik und Beispiele, reguläre Ausdrücke und reguläre Sprachen, reguläre Ausdrücke und endliche Automaten</li> <li>4. Kellerautomaten Syntax, Semantik und Beispiele, deterministische/nichtdeterministische Kellerautomaten</li> <li>5. Grammatiken Syntax, Semantik und Beispiele, Grammatiken und die Chomsky-Hierarchie, Grammatiken und Sprachen, Mehrdeutigkeit, Anwendung: Z.B. XML, DTD, Typschemata und Validierung</li> <li>6. Turingmächtigkeit Syntax, Semantik und Beispiele, berechenbare und nicht-berechenbare Funktionen, Halteproblem</li> </ol> <p>In der Übung werden Themen des seminaristischen Unterrichts vertieft.</p>
<b>Literatur</b>	<p>J. E. Hopcroft, R. Motwani, J. D. Ullman: <i>Introduction to Automata Theory, Languages and Computation</i>, Addison-Wesley Publishing Company</p> <p>U. Hedtstück: <i>Einführung in die Theoretische Informatik – Formale Sprachen und Automatentheorie</i>, Oldenbourg Verlag</p>
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>B03</b>
<b>Titel</b>	<b>Mediendesign Grundlagen / Principles of Media Design</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (2 SWS SU + 2 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	<p>Die Studierenden erwerben grundlegende Kenntnisse zur plattformübergreifenden Gestaltung digitaler und interaktiver Anwendungen und erlernen den praxisorientierten Umgang mit digitalen Medien.</p> <p>Die Studierenden sollen in der Lage sein, Designkonzepte für verschiedene Medienarten und mit unterschiedlichen Anforderungen an das Interface zu entwickeln: vom Small Screen Design am Smartphone bis zur Webanwendung am Computer.</p> <p>Anhand einer konkreten Themenstellung, die das Semester wie ein roter Faden durchzieht, werden wichtige Aspekte der Gestaltung und Konzeption von digitalen Medien erlernt. Der Fokus liegt dabei auf dem gestalterischen Experimentieren mit digitalen Werkzeugen in Kombination mit einer strukturierten, methodischen Vorgehensweise.</p> <p>Die Studierenden sind in der Lage digitale, interaktive Anwendungen zu konzipieren, zu visualisieren und zu erproben.</p>
<b>Voraussetzungen</b>	keine
<b>Niveaustufe</b>	1. Studienplansemester
<b>Lehrform</b>	Seminaristischer Unterricht, Übung
<b>Status</b>	Pflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 75% gestalterische Übungen (14-tägige Übungsblätter), 25% Projektarbeit (20 Stunden).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<ul style="list-style-type: none"> <li>• Designgrundlagen für elektronische und interaktive Medien (Screen- und Interfacedesign für verschiedene mobile und interaktive Medien)</li> <li>• Digitaler Entwurf: grafische Grundbausteine, Informationsarchitektur, Gestaltungsraster</li> <li>• Visuelle Zeichen im GUI (Icons, Symbole, Logos und Markenzeichen etc.)</li> <li>• Typografie: Schriftenklassifikation, Schriftgestaltung, Schriften digital</li> <li>• Farbe: Farbmodelle, Farbkombination, Farbkontraste</li> <li>• Bilder digital (Bildwelten, Bildbearbeitung, Bildkommunikation)</li> <li>• Layout: (Formate, Gestaltungsprinzipien, Medienadäquanz der Gestaltung)</li> <li>• Repertoirebildung an digitalen Tools und Gestaltungsmitteln</li> <li>• Neue Entwicklungen und Innovationen im Mediendesign</li> </ul>
<b>Literatur</b>	<p>T. Stapelkamp: <i>Screen- und Interfacedesign. Gestaltung und Usability für Hard- und Software</i>, Springer Verlag</p> <p>C. Moser: <i>User Experience Design: Mit erlebniszentrierter Softwareentwicklung zu Produkten, die begeistern</i>, Springer Verlag; X.media.press</p> <p>M. Spies: <i>Branded Interactions: Digitale Markenerlebnisse planen und gestalten</i>, Schmidt Hermann Verlag</p>
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.



<b>Modulnummer</b>	<b>B04</b>
<b>Titel</b>	<b>Technische Grundlagen der Informatik / Technical Principles of Computer Science</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (2 SWS SU + 2 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden lernen den grundsätzlichen Aufbau eines Rechners, die Darstellung von Zahlen im Rechner sowie die Bedeutung der verschiedenen Systemkomponenten kennen.
<b>Voraussetzungen</b>	Empfehlung: Beherrschung des Umgangs mit einem Rechner/Standardsoftware
<b>Niveaustufe</b>	1. Studienplansemester
<b>Lehrform</b>	Seminaristischer Unterricht, Übung
<b>Status</b>	Pflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 100% Klausur (90 Minuten).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<ol style="list-style-type: none"> <li>1. Information und Nachricht <ul style="list-style-type: none"> <li>• Digitale Nachrichten, Codierung, Nachrichten- und Informationsverarbeitung</li> </ul> </li> <li>2. Zahlensysteme <ul style="list-style-type: none"> <li>• Die Ursprünge von Zahlensystemen</li> <li>• Stellenwertcodes und Konvertierung ganzer Zahlen</li> <li>• Darstellung negativer ganzer Zahlen</li> <li>• Addition und Subtraktion</li> <li>• Darstellung von Gleitpunktzahlen</li> <li>• Arithmetische Operation für Maschinenzahlen</li> <li>• Grundlagen der Computer Numerik</li> </ul> </li> <li>3. Lokale Grundkonzepte <ul style="list-style-type: none"> <li>• Grundlagen der Boole'schen Algebra</li> <li>• Schaltfunktionen und Schaltnetze</li> <li>• Grundlegende Bausteine (Addierer, Subtrahierer, Dekoder, ...)</li> </ul> </li> <li>4. Rechnerhardware <ul style="list-style-type: none"> <li>• Aufbau eines Rechners</li> <li>• Rechner- und Prozessorarchitekturen</li> <li>• Architektur virtueller Maschinen</li> <li>• Speichertechnologien</li> </ul> </li> <li>5. Mensch-Maschineschnittstelle <ul style="list-style-type: none"> <li>• Ein-/Ausgabemedien</li> <li>• VGA-Schnittstelle, Grafikkbeschleuniger</li> </ul> </li> <li>6. Maßnahmen zur Leistungssteigerung <ul style="list-style-type: none"> <li>• Cache, Pipelines</li> <li>• Multimedia-Erweiterungen</li> </ul> </li> </ol> <p>In der Übung werden Themen des seminaristischen Unterrichts vertieft.</p>
<b>Literatur</b>	W. Oberschelp, G. Vossen: <i>Rechneraufbau und Rechnerstrukturen</i> , Oldenbourg
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>B05</b>
<b>Titel</b>	<b>Programmierung I / Programming 1</b>
<b>Leistungspunkte</b>	10 LP
<b>Workload</b>	136 Stunden Präsenz (4 SWS SU + 4 SWS Ü), 164 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden kennen die grundlegenden Konzepte der objektorientierten Programmierung. Sie entwickeln eine erste Vorstellung von gut strukturierten Klassen und der Interaktion von Objekten. Sie haben einige Klassen aus der Java Standardbibliothek kennen gelernt und sind in der Lage eine Klassendokumentation zu verstehen. Sie sind in der Lage mit Hilfe von eigenen Klassen und grundlegende Klassen aus der Java Standardbibliothek geeignete Lösungen für kleinere Problemstellungen zu entwickeln. Dabei achten sie auf gut strukturierten und lesbaren Code.
<b>Voraussetzungen</b>	Empfehlung: Beherrschung des Umgangs mit einem Rechner und Standardsoftware
<b>Niveaustufe</b>	1. Studienplansemester
<b>Lehrform</b>	Seminaristischer Unterricht
<b>Status</b>	Pflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 50% Klausur (90 Minuten), 50% schriftliche Übungen (14-tägige Übungsblätter).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<ul style="list-style-type: none"> <li>• Aufbau von Klassen (Attribute, Konstruktoren, Methoden)</li> <li>• Variablen und Datentypen (elementare Datentypen, Referenztypen)</li> <li>• Anweisungen (Zuweisung, Bedingte Anweisungen, Schleifen)</li> <li>• Code Konventionen und Programmierstil</li> <li>• Objektinteraktion (interner/externer Methodenaufruf)</li> <li>• Objektsammlungen (z.B. Arrays, Listen, Mengen, Iteratoren)</li> <li>• Kapselung von Klassen (Schnittstelle, Geheimnisprinzip)</li> <li>• Nutzung der Standardbibliothek</li> <li>• Dokumentation von Klassen</li> <li>• Klassenattribute und -methoden</li> <li>• Vererbung/Polymorphie</li> </ul> <p>In der Übung werden Themen des seminaristischen Unterrichts am Rechner vertieft.</p>
<b>Literatur</b>	<p>D. J. Barnes, M. Kölling: <i>Java lernen mit BlueJ : Eine Einführung in die objektorientierte Programmierung</i>, Pearson Studium</p> <p>K. Sierra, B. Bates: <i>Java von Kopf bis Fuß</i>, O'Reilley</p> <p>H.-P. Habelitz: <i>Programmieren Lernen mit Java</i>, Rheinwerk Computing</p> <p>H. Mössenböck: <i>Sprechen Sie Java?</i>, dpunkt.verlag</p> <p>P. Pepper: <i>Programmieren mit Java</i>, Springer Verlag</p>
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>B06</b>
<b>Titel</b>	<b>Mathematik II / Mathematics 2</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	102 Stunden Präsenz (4 SWS SU + 2 SWS Ü), 48 Stunden Selbststudium
<b>Lerngebiet</b>	Mathematisch-naturwissenschaftliche Grundlagen
<b>Lernziele / Kompetenzen</b>	Erkennen von Zusammenhängen und ihrer mathematischen Formulierung, sichere Handhabung von Lösungstechniken dafür.
<b>Voraussetzungen</b>	Empfehlung: Mathematik I
<b>Niveaustufe</b>	2. Studienplansemester
<b>Lehrform</b>	Seminaristischer Unterricht, Übung
<b>Status</b>	Pflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 100% Klausur (90 Minuten).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<p>II Lineare Algebra</p> <ul style="list-style-type: none"> <li>• Lineare Gleichungssysteme und Gaußsches Eliminationsverfahren</li> <li>• Matrixalgebra: Multiplikation, Inverse, Determinanten</li> <li>• Vektoralgebra: Vektoren und Skalare, Rechenoperationen und Gesetze, Linearkombinationen und Koordinatensysteme, lineare Unabhängigkeit, Basis</li> <li>• Analytische Geometrie: Längen und Winkel, Geraden und Ebenen, orthogonale Projektion</li> </ul> <p>III Elemente der Analysis</p> <ul style="list-style-type: none"> <li>• Elementare Funktionen: Potenzen/Wurzeln, rationale Funktionen, Exponential- und Logarithmusfunktionen, Winkelfunktionen, Betrag, Signum, Rundung, Modulo</li> <li>• Grundlegende Eigenschaften der elem. Funktionen: Monotonie, Symmetrie, Nullstellen, Umkehrfunktion</li> <li>• Zahlenfolgen und Konvergenz</li> </ul>
<b>Literatur</b>	G. + S. Teschl: <i>Mathematik für Informatiker Band 1</i> , Springer+Vieweg Verlag P. Hartmann: <i>Mathematik für Informatiker</i> , Vieweg+Teubner Verlag H. Anton: <i>Lineare Algebra</i> , Spektrum Verlag
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>B07</b>
<b>Titel</b>	<b>Algorithmen und Datenstrukturen / Algorithms and Data Structures</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (2 SWS SU + 2 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden sollen Modellierungstechniken, insbesondere Graphen, grundlegende Datenstrukturen sowie Algorithmen zur Lösung von Problemen kennenlernen. Sie können Laufzeiten von Algorithmen abschätzen und effiziente Algorithmen unter Verwendung geeigneter Datenstrukturen entwerfen und programmieren.
<b>Voraussetzungen</b>	Empfehlung: Grundlegende mathematische und Programmierkenntnisse
<b>Niveaustufe</b>	2. Studienplansemester
<b>Lehrform</b>	Seminaristischer Unterricht, Übung
<b>Status</b>	Pflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 75% Klausur (90 Minuten), 25% schriftliche Übungen (14-tägige Übungsblätter).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<ol style="list-style-type: none"> <li>1. Komplexitätsmaße und O-Notation</li> <li>2. Polynomialzeit, NP-Vollständigkeit</li> <li>3. Datenstrukturen, wie z. B.: Arrays, Listen, Bäume, Heaps, Hashtabellen</li> <li>4. Elementare Algorithmen, wie z.B. Sortieralgorithmen, Hashverfahren, Durchlaufverfahren auf Bäumen</li> <li>5. Graphen und Graphenalgorithmen: Breitensuche, Tiefensuche, Kürzeste-Wege-Algorithmen, Bestimmung minimal spannender Bäume, Bestimmung von Matchings, Bestimmung von Flüssen in Netzwerken</li> <li>6. Optional: Allgemeine Algorithmische Verfahren, wie beispielsweise Greedy-Algorithmen, Divide-And-Conquer-Algorithmen, Dynamische Programmierung, Lineare Programmierung, Probabilistische Algorithmen</li> </ol> <p>In der Übung implementieren die Studierenden einige der Datenstrukturen und Algorithmen und erlernen das Profiling, die Ausführung von Algorithmen „mit der Hand“.</p>
<b>Literatur</b>	T. H. Cormen u.a.: <i>Algorithmen – Eine Einführung</i> , Oldenbourg T. Ottmann, P. Widmayer: <i>Algorithmen und Datenstrukturen</i> , Spektrum Akademischer Verlag
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>B08</b>
<b>Titel</b>	<b>Datenbanksysteme / Database Systems</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (2 SWS SU + 2 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden verstehen Aufbau und Wirkungsweise von Datenbanksystemen. Sie sind in der Lage, einen kompletten Datenbankentwurf durchzuführen, Datenbanken zu erstellen, Datenbanken interaktiv und aus Anwendungsprogrammen anzufragen und zu manipulieren.
<b>Voraussetzungen</b>	Empfehlung: Programmierkenntnisse
<b>Niveaustufe</b>	2. Studienplansemester
<b>Lehrform</b>	Seminaristischer Unterricht, Übung
<b>Status</b>	Pflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 75% Klausur (90 Minuten), 25% schriftliche Übungen (14-tägige Übungsblätter).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<ol style="list-style-type: none"> <li>1. Datenbanken – Grundlagen und Einführung</li> <li>2. Datenanalyse, Datenmodellierung und DB-Entwurf <ul style="list-style-type: none"> <li>• Datenmodelle, z.B. Relationales Datenmodell, ER-Modell</li> <li>• Phasenmodell des Datenbankentwurfs</li> <li>• Abbildung des ER-Modells auf das Relationale Datenmodell</li> <li>• Normalisierung</li> <li>• Relationenalgebra</li> </ul> </li> <li>3. Structured Query Language (SQL) <ul style="list-style-type: none"> <li>• Datendefinition, Datenbankabfragen, Datenmanipulation, Transaktionssteuerung</li> </ul> </li> <li>4. Sichten, Rechtevergabe, Integritätssicherung (Trigger)</li> <li>5. Transaktionsmanagement und Synchronisation</li> <li>6. Datenbank-Anwendungen <ul style="list-style-type: none"> <li>• Architekturansätze</li> <li>• Datenmanipulation in Anwendungsprogrammen</li> </ul> </li> <li>7. Neue Datenbankentwicklungen, z.B. Shared Nothing / Shared Memory Architekturen, NoSQL-DB, Datenbanken für das Semantic Web (RDF, OWL, SPARQL), Geodatenbanken</li> </ol> <p>In der Übung werden die Inhalte des seminaristischen Unterrichts am Rechner vertieft.</p>
<b>Literatur</b>	<p>G. Saake, K. Sattler, A. Heuer: <i>Datenbanken: Konzepte und Sprachen</i>, MITP-Verlag</p> <p>Kemper, Eickler: <i>Datenbanksysteme</i>, Oldenbourg</p> <p>T. Kudraß (Hrsg.): <i>Taschenbuch Datenbanken</i>, Hanser-Verlag</p> <p>G. Vossen: <i>Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme</i>, Oldenbourg</p>
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>B09</b>
<b>Titel</b>	<b>Programmierung II / Programming 2</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (2 SWS SU + 2 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden lernen umfangreiche Bibliotheken einer objektorientierten Programmiersprache eigenständig zu erarbeiten und anzuwenden. Sie sind in der Lage eine einfache Anwendung in Java zu erstellen und diese mit einer geeigneten grafischen Benutzeroberflächen zu versehen.
<b>Voraussetzungen</b>	Empfehlung: Programmierung I
<b>Niveaustufe</b>	2. Studienplansemester
<b>Lehrform</b>	Seminaristischer Unterricht, Übung
<b>Status</b>	Pflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 100% Klausur (90 Minuten).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<ul style="list-style-type: none"> <li>• Abstrakte Klassen</li> <li>• Interfaces</li> <li>• Ausnahmebehandlung</li> <li>• Testen (mit JUnit) / Debugging</li> <li>• Vertiefung Objektsammlungen (z.B. Map, Sortierte Sammlungen)</li> <li>• Aufbau grafischer Benutzeroberflächen</li> <li>• Ereignisbehandlung</li> <li>• Serialisierung</li> <li>• Datei Ein-/Ausgabe</li> <li>• Optional: Weitere vertiefende Themen oder APIs</li> </ul> <p>In der Übung entwickeln die Studierenden eine komplexere Anwendung, in der die vorgestellten Konzepte praktisch eingesetzt werden.</p>
<b>Literatur</b>	<p>K. Sierra, B. Bates: <i>Java von Kopf bis Fuß</i>, O'Reilley</p> <p>C. Ullenboom: <i>Java ist auch eine Insel</i>, Galileo Computing – online verfügbar</p> <p>C. S. Horstmann, G. Cornell: <i>Core Java, Band 1 – Grundlagen</i>, Addison-Wesley</p> <p>H.-P. Habelitz: <i>Programmieren Lernen mit Java</i>, Rheinwerk Computing</p> <p>H. Mössenböck: <i>Sprechen Sie Java?</i>, dpunkt.verlag</p> <p>P. Pepper: <i>Programmieren mit Java</i>, Springer Verlag</p>
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>B10</b>
<b>Titel</b>	<b>Betriebssysteme / Operating Systems</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (2 SWS SU + 2 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden verstehen Aufbau und Wirkungsweise von Betriebssystemen, speziell von Mehrbenutzersystemen.
<b>Voraussetzungen</b>	Empfehlung: Programmierkenntnisse
<b>Niveaustufe</b>	2. Studienplansemester
<b>Lehrform</b>	Seminaristischer Unterricht, Übung
<b>Status</b>	Pflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 100% Klausur (90 Minuten).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<ul style="list-style-type: none"> <li>• Aufgaben eines Betriebssystems</li> <li>• Unix Shell</li> <li>• Prozessverwaltung</li> <li>• Dateisysteme</li> <li>• Sicherheitstechniken</li> </ul> <p>In den Übungen erfolgt eine Vertiefung der Inhalte des seminaristischen Unterrichts am Rechner</p> <ul style="list-style-type: none"> <li>• Arbeiten mit der UNIX-Standardshell sh</li> <li>• Moderne Skriptsprachen zur Systemverwaltung (z.B. Python)</li> </ul>
<b>Literatur</b>	A. Tanenbaum: <i>Moderne Betriebssysteme</i> , Pearson Studium
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>B11</b>
<b>Titel</b>	<b>Studium Generale I / General Studies 1</b>
<b>Leistungspunkte</b>	2,5 LP
<b>Workload</b>	34 Stunden Präsenz (2 SWS SU oder 2 SWS Ü), 41 Stunden Selbststudium
<b>Lerngebiet</b>	Allgemeinwissenschaftliche Ergänzungen
<b>Lernziele / Kompetenzen</b>	Die fachübergreifenden Lehrinhalte dienen der interdisziplinären Erweiterung des Fachstudiums und dem Erkennen von Zusammenhängen zwischen Gesellschaft und ihren Teilsystemen.
<b>Voraussetzungen</b>	keine (Ausnahmen können für die Fremdsprachen festgelegt werden)
<b>Niveaustufe</b>	Bachelor- und Masterstudiengänge
<b>Lehrform</b>	Seminaristischer Unterricht, Übungen, Referate, Rollenspiele, Textarbeit, ..... je nach gewähltem Modul
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	siehe Beschreibung der jeweiligen Lehrveranstaltung
<b>Ermittlung der Modulnote</b>	siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<p>In den ingenieur- und naturwissenschaftlichen Studiengängen sind dazu Lerninhalte aus den Bereichen:</p> <ul style="list-style-type: none"> <li>• Politik- und Sozialwissenschaften</li> <li>• Geisteswissenschaften</li> <li>• Wirtschafts-, Rechts- und Arbeitswissenschaften</li> <li>• Fremdsprachen</li> </ul> <p>zu berücksichtigen.</p> <p>In den wirtschaftswissenschaftlichen Studiengängen sind jeweils Lerninhalte aus den Bereichen:</p> <ul style="list-style-type: none"> <li>• Politik- und Sozialwissenschaften</li> <li>• Geisteswissenschaften</li> <li>• Natur- und Ingenieurwissenschaften</li> <li>• Fremdsprachen</li> </ul> <p>zu berücksichtigen.</p>
<b>Literatur</b>	Wird in den jeweiligen Beschreibungen der Lehrveranstaltungen angegeben
<b>Weitere Hinweise</b>	Die Auswahl der Lehrveranstaltungen dieses Moduls obliegt der Eigenverantwortung der Studierenden. Die Auswahl der Lehrveranstaltungen müssen die Studierenden aus den für ihren Studiengang zugelassenen Bereichen treffen (siehe Inhalt)



<b>Modulnummer</b>	<b>B12</b>
<b>Titel</b>	<b>Studium Generale II / General Studies 2</b>
<b>Leistungspunkte</b>	2,5 LP
<b>Workload</b>	34 Stunden Präsenz (2 SWS SU oder 2 SWS Ü), 41 Stunden Selbststudium
<b>Lerngebiet</b>	Allgemeinwissenschaftliche Ergänzungen
<b>Lernziele / Kompetenzen</b>	Die fachübergreifenden Lehrinhalte dienen der interdisziplinären Erweiterung des Fachstudiums und dem Erkennen von Zusammenhängen zwischen Gesellschaft und ihren Teilsystemen.
<b>Voraussetzungen</b>	keine (Ausnahmen können für die Fremdsprachen festgelegt werden)
<b>Niveaustufe</b>	Bachelor- und Masterstudiengänge
<b>Lehrform</b>	Seminaristischer Unterricht, Übungen, Referate, Rollenspiele, Textarbeit, ..... je nach gewähltem Modul
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	siehe Beschreibung der jeweiligen Lehrveranstaltung
<b>Ermittlung der Modulnote</b>	siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<p>In den ingenieur- und naturwissenschaftlichen Studiengängen sind dazu Lerninhalte aus den Bereichen:</p> <ul style="list-style-type: none"> <li>• Politik- und Sozialwissenschaften</li> <li>• Geisteswissenschaften</li> <li>• Wirtschafts-, Rechts- und Arbeitswissenschaften</li> <li>• Fremdsprachen</li> </ul> <p>zu berücksichtigen.</p> <p>In den wirtschaftswissenschaftlichen Studiengängen sind jeweils Lerninhalte aus den Bereichen:</p> <ul style="list-style-type: none"> <li>• Politik- und Sozialwissenschaften</li> <li>• Geisteswissenschaften</li> <li>• Natur- und Ingenieurwissenschaften</li> <li>• Fremdsprachen</li> </ul> <p>zu berücksichtigen.</p>
<b>Literatur</b>	Wird in den jeweiligen Beschreibungen der Lehrveranstaltungen angegeben
<b>Weitere Hinweise</b>	Die Auswahl der Lehrveranstaltungen dieses Moduls obliegt der Eigenverantwortung der Studierenden. Die Auswahl der Lehrveranstaltungen müssen die Studierenden aus den für ihren Studiengang zugelassenen Bereichen treffen (siehe Inhalt)

<b>Modulnummer</b>	<b>B13</b>
Titel	<b>Software Engineering I / Software Engineering 1</b>
Leistungspunkte	5 LP
Workload	68 Stunden Präsenz (2 SWS SU + 2 SWS Ü), 82 Stunden Selbststudium
Lerngebiet	Fachspezifische Grundlagen
Lernziele / Kompetenzen	Die Studierenden sollen in der Lage sein, Software-Anwendungen eigenständig zu konzipieren. Die zur Anwendung kommende Modellierungssprache (z.B. UML) soll von den Studierenden beherrscht werden. Die Kompetenz besteht in der ingenieurmäßigen Konzeption von SW-Produkten (Schwerpunkt OOA). Die Studierenden erkennen die Relevanz der Systemanalyse. Weiterhin erwerben sie Kenntnisse und Fertigkeiten im Bereich Soft-Skills, die bei der konzeptionellen Arbeit für Software-Systeme wichtig sind.
Voraussetzungen	Empfehlung: Kenntnisse einer objektorientierten Programmiersprache
Niveaustufe	3. Studienplansemester
Lehrform	Seminaristischer Unterricht, Übung
Status	Pflichtmodul
Häufigkeit des Angebotes	jedes Semester
Prüfungsform	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 75% Klausur (90 Minuten), 25% schriftliche Übungen (14-tägige Übungsblätter).
Ermittlung der Modulnote	Siehe Studienplan
Anerkannte Module	Module vergleichbaren Inhalts
Inhalte	<ol style="list-style-type: none"> <li>1. Software-Entwicklung: Einführung <ul style="list-style-type: none"> <li>• Paradigmen der Software-Entwicklung und Software-Entwicklungsmethoden in Verbindung mit Vorgehensmodellen</li> <li>• Software-Entwicklung im Kontext des Konfigurations- und Qualitätsmanagement</li> </ul> </li> <li>2. Einführung in das Requirements Engineering/Anforderungsmanagement <ul style="list-style-type: none"> <li>• Requirements Engineering und Systemanalyse: Ziele und Methoden</li> <li>• Erstellung einer Anforderungsspezifikation, Lastenheft</li> </ul> </li> <li>3. Systemanalyse/Sollkonzept <ul style="list-style-type: none"> <li>• Übergang vom Lastenheft zum Pflichtenheft, OOA</li> <li>• Geschäftsprozess- und Klassenmodellierung mit der UML (Aktivitätsdiagramm, Zustandsdiagramm, Paket- und Klassendiagramm)</li> <li>• Geschäftsregeln mit OCL</li> </ul> </li> <li>4. Qualitätssicherung der Ergebnisse der Analysephase <ul style="list-style-type: none"> <li>• Abnahme</li> <li>• Qualitätssicherung des Pflichtenheftes, Prüfung der Ergebnisse der Analysephase</li> </ul> </li> </ol> <p>In der Übung erlernen die Studierenden die Systemanalyse, Erstellung eines Pflichtenheftes, Test durch Darstellung des statischen Modells mit Fachklassendiagramm. Anwendung entsprechender Werkzeuge: Requirements-Engineering, UML/OCL-Modellierung.</p>
Literatur	<p>M. Winter: <i>Methodische objektorientierte Softwareentwicklung</i>, dpunkt.verlag</p> <p>M. Hitz, G. Kappel: <i>UML@work</i>, dpunkt.verlag</p> <p>H. Balzert: <i>Lehrbuch der Objektmodellierung – Analyse und Entwurf mit der UML2</i>, Spektrum Verlag</p>
Weitere Hinweise	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>B14</b>
<b>Titel</b>	<b>Computergrafik Grundlagen / Principles of Computer Graphics</b>
<b>Leistungspunkte</b>	7 LP
<b>Workload</b>	68 Stunden Präsenz (2 SWS SU + 2 SWS Ü), 142 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden lernen die Grundlagen der digitalen Bildsynthese kennen. Sie lernen einfache Verfahren zur Bilderzeugung zusammenhängend zu implementieren und anzuwenden.
<b>Voraussetzungen</b>	Empfehlung: Neben den Modulen Mathematik I + II wird aufgrund des hohen praktischen Implementierungsanteils in den Übungen und den Hausarbeiten der erfolgreiche Abschluss der Module Programmieren I + II dringend empfohlen.
<b>Niveaustufe</b>	3. Studienplansemester
<b>Lehrform</b>	Seminaristischer Unterricht, Programmierübung, Hausarbeit
<b>Status</b>	Pflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 50% Klausur (90 Minuten), 50% schriftliche Übungen (14-tägige Übungsblätter).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<p>Das Modul führt in die Grundlagen der digitalen Bildsynthese ein, und behandelt im seminaristischen Unterricht folgende Themen:</p> <ul style="list-style-type: none"> <li>• Trigonometrie, Vektoren, Punkte, Normalen, Oberflächen</li> <li>• Koordinatensysteme und Transformationen</li> <li>• Licht, Farbe und Radiometrie</li> <li>• Sampling und Aliasing</li> <li>• Modellierung und Geometrie</li> <li>• Rendering und Ray-Tracing</li> <li>• Kameramodelle</li> <li>• Beleuchtung, Schattierung und Reflexion</li> <li>• Material und Textur</li> </ul> <p>In den Übungen werden die vermittelten Techniken zur Bilderzeugung aufbauend auf den bisher gesammelten Programmiererfahrungen von Grund auf implementiert.</p>
<b>Literatur</b>	<p>M. Pharr, G. Humphreys. <i>Physically Based Rendering, Second Edition: From Theory To Implementation</i>. Morgan Kaufmann.</p> <p>K. Suffern. <i>Ray Tracing from the Ground Up</i>. A K Peters</p>
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>B15</b>
<b>Titel</b>	<b>Medientechnologien / Media Technologies</b>
<b>Leistungspunkte</b>	8 LP
<b>Workload</b>	102 Stunden Präsenz (2 SWS SU + 4 SWS Ü), 138 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden kennen die wichtigsten Eigenschaften des Sehsinns und des Gehörsinns. Sie kennen die Grundlagen der Lichtausbreitung, Optik, Akustik, Wandlung analoger Signale, der Aufzeichnung und die daraus resultierenden digitalen Formate. Sie haben Kenntnisse über die Grundlagen der Signalverarbeitung. Sie kennen Prinzipien der Datenreduktion und Kompression und der Grundzüge der Signalverteilung in Systemen und in den Distributionskanälen.
<b>Voraussetzungen</b>	Empfehlung: Grundlegende mathematische Kenntnisse
<b>Niveaustufe</b>	3. Studienplansemester
<b>Lehrform</b>	Seminaristischer Unterricht, Übung
<b>Status</b>	Pflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 65% Klausur (90 Minuten), 35% schriftliche Übungen (14-tägige Übungsblätter).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<ul style="list-style-type: none"> <li>• Physikalische und biologische Grundlagen: Licht und Farbe, Optik, Schallwellen, Akustik und visuelle Wahrnehmung,</li> <li>• Messtechnik Grundlagen: Fourieranalyse, Zeit und Frequenzbereich, akustische Wahrnehmung, Waveform, Vectorscope</li> <li>• Grundlagen der Signalverarbeitung: Elektro-mechanische Wandler, A/D-Wandler, Bildabtastung, analoge und digitale Signalformate, Filter, Speicherung, Übertragung</li> <li>• Grundlagen der Audio-, Bild- und Videokompression: MP3, H.264, JPEG (DCT, Huffman-Codierung), MPEG4</li> <li>• Grundlagen zu Codes und Formaten in Systemen und Distributionen</li> </ul> <p>Übungen zu Grundlagen mit ausgewählten Werkzeugen zur Messtechnik, Signalverarbeitung und Kompression.</p>
<b>Literatur</b>	<p>U. Schmidt: <i>Professionelle Videotechnik</i>, Springer Berlin Heidelberg</p> <p>T. Strutz: <i>Bilddatenkompression. Grundlagen, Codierung, JPEG, MPEG, H264</i>, Vieweg-Verlag</p> <p>T. Petrasch, J. Zinke: <i>Videofilmproduktion</i>, Carl Hanser Verlag</p> <p>M. Dickreiter: <i>Handbuch der Tonstudioteknik</i>, De Gruyter SAUR-Verlag</p> <p>H. Henle: <i>Das Tonstudio Handbuch</i>, GC Carstensen Verlag</p> <p>D. Stotz: <i>Computergestützte Audio- und Videotechnik; Multimedialechnik in der Anwendung</i>, Springer Verlag</p>
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>B16</b>
<b>Titel</b>	<b>Verteilte Systeme / Distributed Systems</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (2 SWS SU + 2 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden sollen die Verteilung von Aufgaben und die Kommunikation in Netzwerken verstehen. Die aktuellen Protokolle sollen soweit beherrscht werden, dass die Studierenden zur Programmierung Verteilter Anwendungen fähig sind.
<b>Voraussetzungen</b>	Empfehlung: Beherrschung der Grundlagen der prozeduralen oder objektorientierten Programmierung
<b>Niveaustufe</b>	3. Studienplansemester
<b>Lehrform</b>	Seminaristischer Unterricht, Übung
<b>Status</b>	Pflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 100% Klausur (90 Minuten).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<ol style="list-style-type: none"> <li>1. Modelle der Kommunikation, Grundlagen und Architekturen Verteilter Systeme</li> <li>2. Transportprotokolle und ihre Programmierschnittstellen</li> <li>3. Grundlagen der IP-Protokolle (IPv4 und IPv6)</li> <li>4. Namensdienste (DNS)</li> <li>5. Medienorientierte Protokolle (SIP, RTP, RTSP, ...)</li> <li>6. Sicherheit</li> <li>7. Middleware</li> </ol> <p>In der Übung:</p> <p>In den Übungen programmieren die Studierenden einfache Verteilte Systeme unterschiedlicher Architekturen (Peer-to-Peer, Client/Server) und Kommunikationsformen (verbindungslos, verbindungsorientiert, messaging) und erlernen die Analyse des Datenverkehrs, z.B. mit WireShark.</p>
<b>Literatur</b>	W. R. Stevens: <i>TCP/IP</i> , Hüthig Telekommunikation Online-Dokumentationen und Standards, z.B. auf <a href="http://www.ietf.org">www.ietf.org</a> und <a href="http://www.iana.org">www.iana.org</a>
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>B17</b>
<b>Titel</b>	<b>Web Engineering I / Web Engineering 1</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (2 SWS SU + 2 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	<p>Die Studierenden kennen grundlegende Technologien für multimediale Web-Applikationen. Sie sind in der Lage sowohl eine klassische Website als auch eine Single Page Application in Teamarbeit systematisch zu planen und umzusetzen.</p> <p>Sie kennen die wichtigsten Qualitätskriterien, die bei der Beurteilung von Webapplikationen zu beachten sind.</p> <p>Sie können ausgewählte Auszeichnungs- und Scriptsprachen praktisch einsetzen.</p> <p>Sie haben Erfahrung mit der Konzeption und Realisation von zeitgemäßen Web-Applikationen mit einer geringen serverseitigen Komplexität.</p>
<b>Voraussetzungen</b>	Empfehlung: Grundlegende Programmierkenntnisse, wie sie in den Studienplansemestern 1 und 2 gelehrt werden.
<b>Niveaustufe</b>	3. Studienplansemester
<b>Lehrform</b>	Seminaristischer Unterricht, Übung
<b>Status</b>	Pflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 50% Klausur (90 Minuten), 50% Projektarbeit (40 Stunden).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<ul style="list-style-type: none"> <li>• Grundlegende Webtechnologien</li> <li>• Architektur von Webapplikationen</li> <li>• einfache ergänzende Bibliotheken und Frameworks</li> <li>• fundamentale Designpattern bei Webapplikationen</li> <li>• Datenaustauschformate</li> <li>• Grundlagen der responsiven Gestaltung von Webapplikationen</li> <li>• Maßnahmen für Plattformunabhängigkeit</li> <li>• Grundlagen der Ereignis-Programmierung</li> <li>• Strategische Fehlersuche</li> <li>• Typische Workflows und Maßnahmen zur Qualitätssicherung</li> </ul> <p>In den Übungen werden ausgewählte Themen des seminaristischen Unterrichts anhand von Aufgaben oder eines Projekts praktisch erfahren und vertieft.</p>
<b>Literatur</b>	Die Literatur sowie die zu empfehlenden Internet-Ressourcen werden semesteraktuell von der Lehrkraft mitgeteilt.
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten. Teile der Literatur, Dokumentation und der zu verwendenden Quellen sind nur in Englisch verfügbar.

<b>Modulnummer</b>	<b>B18</b>
<b>Titel</b>	<b>Software Engineering II / Software Engineering 2</b>
<b>Leistungspunkte</b>	8 LP
<b>Workload</b>	102 Stunden Präsenz (2 SWS SU + 4 SWS Ü), 138 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Vertiefung
<b>Lernziele / Kompetenzen</b>	Die Studierenden lernen interaktive Software-Anwendungen auf der Grundlage einer Systemanalyse eigenständig zu entwerfen und zu implementieren. Dabei wenden Sie ausgewählte Architekturmuster und Entwurfskonzepte auf ein konkretes Beispiel an und vertiefen ihr Verständnis für das Zusammenspiel bereits bekannter und neuer Konzepte der objektorientierten Programmierung. Sie bauen wichtige Kompetenzen im Bereich Softwareentwurf und Software-Architektur (OOD) auf. Für die Umsetzung ihrer Entwürfe lernen die Studierenden aktuelle Modellierungs- und Programmiersprachen sowie relevante Entwicklungs-Werkzeuge und -Bibliotheken gezielt einzusetzen.
<b>Voraussetzungen</b>	Empfehlung: Software-Engineering I
<b>Niveaustufe</b>	4. Studienplansemester
<b>Lehrform</b>	Seminaristischer Unterricht, Übung
<b>Status</b>	Pflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 50% Klausur (90 Minuten), 50% Projektarbeit (40 Stunden).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<p>1. Software-Architekturen und Software-Entwurf (OOD):</p> <ul style="list-style-type: none"> <li>• SW-Architekturen für verteilte und multimediale Anwendungen (Multi-Tier-Arch.)</li> <li>• Frameworks</li> <li>• Ebenen, Komponenten</li> <li>• Entwurfsprinzipien und -methoden, Entwurfsmuster</li> <li>• UML-Einsatz im Entwurf (Klassendiagramm, Interaktionsdiagramm, Zustandsdiagramm etc.)</li> <li>• Entwurfskommunikation</li> </ul> <p>2. Implementierung (OOP):</p> <ul style="list-style-type: none"> <li>• Transformation der Entwurfsmodelle</li> <li>• Abbildung der Architektur auf statische und dynamische Code-Bereiche (Pakethierarchien, Klassen, Algorithmen, etc.)</li> <li>• Nutzung von Frameworks und Bibliotheken für die Umsetzung des Entwurfs. z.B. Dependency-Injection-Container, OR-Mapper für die Persistenz</li> </ul> <p>3. Versionierung, Konfigurationsmanagement</p> <p>In der Übung entwickeln die Studierenden, geführt durch konkrete Aufgaben, eine komplexe interaktive Anwendung. Definition und Planung der Entwurfsphase, der Entwurf sowie die Implementierung der Software werden hierbei explizit durchlaufen.</p>
<b>Literatur</b>	G. Starke: <i>Effektive Softwarearchitekturen: Ein praktischer Leitfaden</i> , Carl Hanser Verlag E. Freeman, K. Sierra, B. Bates: <i>Entwurfsmuster von Kopf bis Fuß</i> , O'Reilly H. Balzert: <i>Objektorientierte Programmierung mit Java 5</i> , Spektrum Akademischer Verlag
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>B19</b>
<b>Titel</b>	<b>Web Engineering II / Web Engineering 2</b>
<b>Leistungspunkte</b>	7 LP
<b>Workload</b>	68 Stunden Präsenz (2 SWS SU + 2 SWS Ü), 142 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Vertiefung
<b>Lernziele / Kompetenzen</b>	<p>Die Studierenden können Ihre Erfahrungen aus der Konzeption und Realisation von Webanwendungen mit einer nicht trivialen Server- und Datenbankanbindung wiedergeben. Die Studierenden können eine serverseitige Schnittstelle für moderne Webanwendungen konzipieren und implementieren.</p> <p>Sie haben in Teamarbeit typische Komplexitätsprobleme erfahren und können diese durch den Einsatz von Entwicklungsmethoden, Architektur- und Entwurfsmustern bewältigen.</p>
<b>Voraussetzungen</b>	Empfehlung: Programmieren I + II, Web Engineering I, Software Engineering I
<b>Niveaustufe</b>	4. Studienplansemester
<b>Lehrform</b>	Seminaristischer Unterricht, Übung
<b>Status</b>	Pflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 30% Klausur (90 Minuten), 70% Projektarbeit (40 Stunden).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<ul style="list-style-type: none"> <li>• Client-Server-Architekturen für (mobile) Webanwendungen</li> <li>• Entwicklungsmethoden und -werkzeuge für Webanwendungen (Vertiefung)</li> <li>• Programmiersprachen und -konzepte für Webanwendungen (Vertiefung)</li> <li>• Server-Schnittstellen-Konzeption (bspw. REST-APIs, WebServices)</li> <li>• Anbindung von Schnittstellen (client-seitig)</li> <li>• Modularisierung und Wiederverwendung (server- und client-seitig)</li> <li>• Strategisches Fehlerlösen (Debugging)</li> <li>• Optional: Authentifizierung und Sicherheit in Webanwendungen</li> <li>• Optional: Streaming im Web</li> <li>• Optional: Semantic Web</li> </ul> <p>Im Rahmen des Unterrichtes werden nach Bedarf Fallstudien aus Unternehmen vorgestellt und diskutiert.</p> <p>In der Übung werden die vermittelten Technologien an kleineren Projektaufgaben geübt.</p>
<b>Literatur</b>	Die Literatur sowie die zu empfehlenden Internet-Ressourcen werden semesteraktuell durch die Lehrkraft festgelegt.
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten. Teile der Literatur, Dokumentation und zu verwendenden Quellen sind nur in Englisch verfügbar.



<b>Modulnummer</b>	<b>B20</b>
<b>Titel</b>	<b>Human-Computer Interaction / Human-Computer Interaction</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (2 SWS SU + 2 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden erwerben eine ergonomiebezogene Sicht auf Software-Anwendungen und setzen sich mit Grundlagen der Wahrnehmungs-, Kognitions- und Handlungsprozesse auseinander. Sie lernen die Qualitätsdimensionen „Gebrauchstauglichkeit“ (Usability), Benutzungserfahrung ("User Experience") und „Barrierefreiheit“ (Accessibility) voneinander zu unterscheiden und fachgerecht zu beurteilen. Sie kennen die wichtigsten Gesetze, Normen und Konventionen der Software-Ergonomie in Grundzügen (ISO/EN/DIN sowie WCAG/BITV) und erwerben die Kompetenz zur methodengeleiteten Gestaltung sowie fachgerechten Evaluation von benutzungsorientierten Interaktionsschnittstellen. Sie eignen sich einschlägige Techniken und Werkzeuge auch praktisch an. Außerdem bearbeiten sie Gestaltungsanforderungen für Schnittstellen mobiler Geräte und behandeln innovative Interaktionsformen.
<b>Voraussetzungen</b>	Empfehlung: Kenntnisse in HTML5, CSS sowie Javascript-Grundkenntnisse
<b>Niveaustufe</b>	4. Studienplansemester
<b>Lehrform</b>	Seminaristischer Unterricht, Übung
<b>Status</b>	Pflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 50% Klausur (90 Minuten), 50% schriftliche Übungen (14-tägige Übungsblätter).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<ul style="list-style-type: none"> <li>• Hardware-, Software- und Web-Ergonomie</li> <li>• Qualität interaktiver Anwendungen: Usability, User Experience und Accessibility</li> <li>• Grundlagen der Wahrnehmung und Kognition</li> <li>• Benutzerprofile und Personas</li> <li>• Handlungsmodelle</li> <li>• Normen und Standards für Software Ergonomie, Usability und Accessibility</li> <li>• Design Best Practices</li> <li>• Verfahren und Werkzeuge der Evaluation von Interaktionsschnittstellen</li> </ul> <p>In der Übung lernen die Studierenden anhand praktischer Gestaltungs- und Evaluationsaufgaben die fallweise geeigneten Methoden und Techniken der Software-Ergonomie zu bestimmen und anzuwenden. Sie erwerben eine praktische Kompetenz in den Bereichen: Benutzer- und Aufgabenanalyse, Interaktions- und Oberflächenevaluation, Prototyping.</p>
<b>Literatur</b>	<p>H. Sharp, Y. Rogers, J. Preece: <i>Interaction Design: Beyond human-computer interaction</i>, Wiley</p> <p>M. Herczeg: <i>Software-Ergonomie: Theorien, Modelle und Kriterien für gebrauchstaugliche interaktive Computersysteme</i>, Oldenbourg Verlag</p> <p>J. Nielsen: <i>Designing Web Usability : The Practice of Simplicity</i>, New Riders Publishing</p> <p>J.Nielsen, R. Budiu, <i>Mobile Usability</i>, New Riders Publishing</p> <p>J. E. Helbusch, K. Probersch: <i>Barrierefreiheit verstehen und umsetzen</i>, dpunkt.verlag</p> <p>P. J. Lynch, S. Horton: <i>Web Style Guide: Basic Design Principles for Creating Web Sites</i>, Yale University Press – online verfügbar</p>
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>B21</b>
Titel	<b>Wahlpflichtmodul I / Required-Elective Module 1</b>
Leistungspunkte	5 LP
Workload	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
Lerngebiet	Fachspezifische Vertiefung
Lernziele / Kompetenzen	Siehe Beschreibung der Wahlpflichtmodule
Voraussetzungen	Siehe Beschreibung der Wahlpflichtmodule
Niveaustufe	4. Studienplansemester
Lehrform	Siehe Beschreibung der Wahlpflichtmodule
Status	Wahlpflichtmodul
Häufigkeit des Angebotes	jedes Semester
Prüfungsform	Siehe Beschreibung der Wahlpflichtmodule
Ermittlung der Modulnote	Siehe Beschreibung der Wahlpflichtmodule
Anerkannte Module	Siehe Beschreibung der Wahlpflichtmodule
Inhalte	Ausgewählte Themen je nach gewähltem Modul aus dem Wahlpflichtmodulkatalog Für dieses Wahlpflichtmodul können aus dem Wahlpflichtmodulkatalog die Module WP01 bis WP16 gewählt werden.
Literatur	Siehe Beschreibung der Wahlpflichtmodule
Weitere Hinweise	Auf Beschluss des Fachbereichsrates des Fachbereichs VI können weitere Module als Wahlpflichtmodule vorgesehen werden. Über das Angebot an weiteren Wahlpflichtmodulen entscheidet der Fachbereichsrat jeweils vor Beginn des Semesters. Die/der Studierende kann auf Antrag auch ein Modul aus einem anderen Bachelor-Studiengang als Wahlpflichtmodul im 4. Studienplansemester wählen. Über den Antrag entscheidet der/die Anerkennungsbeauftragte des Fachbereichs. Bei einem zeitweiligen Studium im Ausland können die dort in Modulen erworbenen Leistungspunkte als Wahlpflichtmodule in vollem Umfang anerkannt werden, wenn die Inhalte der Module nicht mit denen der Pflichtmodule dieses Studienplans vergleichbar sind. Über die Anerkennung entscheidet der/die Anerkennungsbeauftragte des Fachbereichs.

<b>Modulnummer</b>	<b>B22</b>
<b>Titel</b>	<b>Wahlpflichtmodul II / Required-Elective Module 2</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Vertiefung
<b>Lernziele / Kompetenzen</b>	Siehe Beschreibung der Wahlpflichtmodule
<b>Voraussetzungen</b>	Siehe Beschreibung der Wahlpflichtmodule
<b>Niveaustufe</b>	4. Studienplansemester
<b>Lehrform</b>	Siehe Beschreibung der Wahlpflichtmodule
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Siehe Beschreibung der Wahlpflichtmodule
<b>Ermittlung der Modulnote</b>	Siehe Beschreibung der Wahlpflichtmodule
<b>Anerkannte Module</b>	Siehe Beschreibung der Wahlpflichtmodule
<b>Inhalte</b>	Ausgewählte Themen je nach gewähltem Modul aus dem Wahlpflichtmodulkatalog Für dieses Wahlpflichtmodul können aus dem Wahlpflichtmodulkatalog die Module WP01 bis WP16 gewählt werden.
<b>Literatur</b>	Siehe Beschreibung der Wahlpflichtmodule
<b>Weitere Hinweise</b>	Auf Beschluss des Fachbereichsrates des Fachbereichs VI können weitere Module als Wahlpflichtmodule vorgesehen werden. Über das Angebot an weiteren Wahlpflichtmodulen entscheidet der Fachbereichsrat jeweils vor Beginn des Semesters. Die/der Studierende kann auf Antrag auch ein Modul aus einem anderen Bachelor-Studiengang als Wahlpflichtmodul im 4. Studienplansemester wählen. Über den Antrag entscheidet der/die Anerkennungsbeauftragte des Fachbereichs. Bei einem zeitweiligen Studium im Ausland können die dort in Modulen erworbenen Leistungspunkte als Wahlpflichtmodule in vollem Umfang anerkannt werden, wenn die Inhalte der Module nicht mit denen der Pflichtmodule dieses Studienplans vergleichbar sind. Über die Anerkennung entscheidet der/die Anerkennungsbeauftragte des Fachbereichs.

<b>Modulnummer</b>	<b>B23</b>
<b>Titel</b>	<b>Projekt / Project</b>
<b>Leistungspunkte</b>	15 LP
<b>Workload</b>	102 Stunden Präsenz (6 SWS Ü), 348 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Vertiefung
<b>Lernziele / Kompetenzen</b>	Die Studierenden sollen in der Lage sein, eine komplexe Anwendung mit projektspezifischen Basistechniken in Teamarbeit zu konzipieren und umzusetzen. Kenntnisse der verwendeten Analyse- und Modellierungsmethoden, Programmier Techniken und Frameworks sowie der relevanten Lehrinhalte der Semester 1 - 4 werden vertieft und mit anderen Themen verknüpft. Teamfähigkeit, Kommunikations- und Präsentationstechniken sowie die Kompetenz zu Konfliktlösung und zu Aufwands- und Risikoabschätzung sollen durch die Projektarbeit gefördert werden.
<b>Voraussetzungen</b>	Empfehlung: Grundlegende Modellierungs- und Programmierkenntnisse
<b>Niveaustufe</b>	5. Studienplansemester
<b>Lehrform</b>	Übung
<b>Status</b>	Pflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 100% Projektarbeit (348 Stunden).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	Im Rahmen des Projekts bearbeiten die Studierenden eine komplexe und aktuelle Aufgabenstellung der Medieninformatik. Die Aufgabe wird von einem Projektteam eigenständig mit projektspezifischer Organisation, Durchführung und Planung bearbeitet.
<b>Literatur</b>	Literatur wird in der Lehrveranstaltung empfohlen.
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>B24</b>
<b>Titel</b>	<b>Grundlagen wissenschaftlichen Arbeitens / Principles of Scientific Work</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	51 Stunden Präsenz (1 SWS SU + 2 SWS Ü), 99 Stunden Selbststudium
<b>Lerngebiet</b>	Fachübergreifende Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden beherrschen die grundlegenden Techniken zum wissenschaftlichen Arbeiten. Sie sind in der Lage Themen zu recherchieren, aufzuarbeiten und Anderen anschaulich zu präsentieren.
<b>Voraussetzungen</b>	keine
<b>Niveaustufe</b>	5. Studienplansemester
<b>Lehrform</b>	Seminaristischer Unterricht, Übung
<b>Status</b>	Pflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 100% Präsentation (20 Minuten).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<p>Einführung in wissenschaftliche Arbeitsweisen</p> <ul style="list-style-type: none"> <li>• Literatursuche und Online-Recherche</li> <li>• Umgang mit wissenschaftlichen Texten</li> <li>• Verfassen von wissenschaftlichen Texten</li> <li>• Anfertigung und Präsentation von Studienleistungen</li> <li>• Inhaltliche und methodische Vorbereitung von Präsentationen</li> <li>• Grundlagen der Rhetorik</li> </ul> <p>In praktischen Übungen werden Fachpräsentationen vorbereitet und gehalten. In Einzelvorträgen erhält jede/r Studierende ein individuelles Feedback zu ihren/seinen persönlichen Stärken und Schwächen.</p>
<b>Literatur</b>	H. Balzert, C. Schäfer, M. Schröder, U. Kern: <i>Wissenschaftliches Arbeiten</i> , W3L Verlag
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>B25</b>
<b>Titel</b>	<b>Wahlpflichtmodul III / Required-Elective Module 3</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Vertiefung
<b>Lernziele / Kompetenzen</b>	Siehe Beschreibung der Wahlpflichtmodule
<b>Voraussetzungen</b>	Siehe Beschreibung der Wahlpflichtmodule
<b>Niveaustufe</b>	5. Studienplansemester
<b>Lehrform</b>	Siehe Beschreibung der Wahlpflichtmodule
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Siehe Beschreibung der Wahlpflichtmodule
<b>Ermittlung der Modulnote</b>	Siehe Beschreibung der Wahlpflichtmodule
<b>Anerkannte Module</b>	Siehe Beschreibung der Wahlpflichtmodule
<b>Inhalte</b>	Ausgewählte Themen je nach gewähltem Modul aus dem Wahlpflichtmodulkatalog Für dieses Wahlpflichtmodul können aus dem Wahlpflichtmodulkatalog die Module WP01 bis WP16 gewählt werden.
<b>Literatur</b>	Siehe Beschreibung der Wahlpflichtmodule
<b>Weitere Hinweise</b>	Auf Beschluss des Fachbereichsrates des Fachbereichs VI können weitere Module als Wahlpflichtmodule vorgesehen werden. Über das Angebot an weiteren Wahlpflichtmodulen entscheidet der Fachbereichsrat jeweils vor Beginn des Semesters. Die/der Studierende kann auf Antrag auch ein Modul aus einem anderen Bachelor-Studiengang als Wahlpflichtmodul im 5. Studienplansemester wählen. Über den Antrag entscheidet der/die Anerkennungsbeauftragte des Fachbereichs. Bei einem zeitweiligen Studium im Ausland können die dort in Modulen erworbenen Leistungspunkte als Wahlpflichtmodule in vollem Umfang anerkannt werden, wenn die Inhalte der Module nicht mit denen der Pflichtmodule dieses Studienplans vergleichbar sind. Über die Anerkennung entscheidet der/die Anerkennungsbeauftragte des Fachbereichs.

<b>Modulnummer</b>	<b>B26</b>
<b>Titel</b>	<b>Wahlpflichtmodul IV / Required-Elective Module 4</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Vertiefung
<b>Lernziele / Kompetenzen</b>	Siehe Beschreibung der Wahlpflichtmodule
<b>Voraussetzungen</b>	Siehe Beschreibung der Wahlpflichtmodule
<b>Niveaustufe</b>	5. Studienplansemester
<b>Lehrform</b>	Siehe Beschreibung der Wahlpflichtmodule
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Siehe Beschreibung der Wahlpflichtmodule
<b>Ermittlung der Modulnote</b>	Siehe Beschreibung der Wahlpflichtmodule
<b>Anerkannte Module</b>	Siehe Beschreibung der Wahlpflichtmodule
<b>Inhalte</b>	Ausgewählte Themen je nach gewähltem Modul aus dem Wahlpflichtmodulkatalog Für dieses Wahlpflichtmodul können aus dem Wahlpflichtmodulkatalog die Module WP01 bis WP16 gewählt werden.
<b>Literatur</b>	Siehe Beschreibung der Wahlpflichtmodule
<b>Weitere Hinweise</b>	Auf Beschluss des Fachbereichsrates des Fachbereichs VI können weitere Module als Wahlpflichtmodule vorgesehen werden. Über das Angebot an weiteren Wahlpflichtmodulen entscheidet der Fachbereichsrat jeweils vor Beginn des Semesters. Die/der Studierende kann auf Antrag auch ein Modul aus einem anderen Bachelor-Studiengang als Wahlpflichtmodul im 5. Studienplansemester wählen. Über den Antrag entscheidet der/die Anerkennungsbeauftragte des Fachbereichs. Bei einem zeitweiligen Studium im Ausland können die dort in Modulen erworbenen Leistungspunkte als Wahlpflichtmodule in vollem Umfang anerkannt werden, wenn die Inhalte der Module nicht mit denen der Pflichtmodule dieses Studienplans vergleichbar sind. Über die Anerkennung entscheidet der/die Anerkennungsbeauftragte des Fachbereichs.

<b>Modulnummer</b>	<b>B27</b>
<b>Titel</b>	<b>Praxisprojekt / Practice-Based Project</b>
Credits	15 LP
Präsenzzeit	0 SWS
Lerngebiet	Fachspezifische Vertiefung
Lernziele / Kompetenzen	Anwendung und Vertiefung der im 1. bis 5. Fachsemester erworbenen Kenntnisse und Fähigkeiten in einem Praxisprojekt. Dazu zählen sowohl fachliche Kompetenzen als auch soziale, kommunikative als auch personale Kompetenzen.
Voraussetzungen	siehe jeweils gültige Ordnung für die Praxisphase
Niveaustufe	6. Studienplansemester
Lehrform	Betriebliche Arbeit
Status	Pflichtmodul
Häufigkeit des Angebotes	jedes Semester
Prüfungsform	Präsentation (30 Minuten), Schriftlicher Bericht (10 – 15 Seiten)
Ermittlung der Modulnote	Siehe Studienplan
Anerkannte Module	Module vergleichbaren Inhalts
Inhalte	Die Inhalte richten sich nach dem jeweiligen Projekt, das in einem Betrieb durchgeführt wird. Eine Vorprüfung der Qualität des Betriebes, sowie der durchzuführenden Aufgaben erfolgt durch die Praxisbeauftragten des Fachbereichs. Den Studierenden wird jeweils ein/e Hochschullehrer/in zugewiesen, der/die die Praxisphase wissenschaftlich begleitet sowie den schriftlichen Bericht und die Präsentation begutachtet.
Literatur	Fachspezifisch, projektabhängig
Weitere Hinweise	Dieses Modul wird auf Deutsch angeboten.



Modulnummer	B28
Titel	<b>Abschlussprüfung / Final Examination Module</b> <b>B28.1 Bachelor-Arbeit / Bachelor's Thesis</b> <b>B28.2 Mündliche Abschlussprüfung / Oral Final Examination</b> <b>(Abschlussarbeit gemäß jeweils geltender Rahmenstudien- und -prüfungsordnung)</b>
Leistungspunkte	15 LP
Präsenzzeit	30 – 45 Minuten mündliche Abschlussprüfung
Lerngebiet	Fachspezifische Vertiefung
Lernziele / Kompetenzen	<u>Bachelor-Arbeit</u> Selbstständige Bearbeitung eines wissenschaftlichen Projektes mit schriftlicher Ausarbeitung ungefähr 50 – 60 Seiten <u>Mündliche Abschlussprüfung</u> Die mündliche Abschlussprüfung orientiert sich schwerpunktmäßig an den Fachgebieten der Abschlussarbeit. Durch die Abschlussprüfung soll festgestellt werden, ob der/die Studierende gesichertes Wissen in den Fachgebieten, denen die Abschlussarbeit thematisch zugeordnet ist, besitzt und fähig ist, die Ergebnisse der Abschlussarbeit selbstständig zu begründen.
Voraussetzungen	Zulassung gemäß geltender Rahmenstudien- und -prüfungsordnung und §7(2) der Studien- und Prüfungsordnung für den Bachelorstudiengang Medieninformatik.
Niveaustufe	6. Studienplansemester
Lehrform	<u>Bachelor-Arbeit</u> Betreute Arbeit; die Betreuung erfolgt gemäß § 29 (7) RSPO durch den/die Betreuer/in der Bachelor-Arbeit <u>Mündliche Abschlussprüfung</u> Präsentation (ca. 15 min) und mündliche Prüfung
Status	Pflichtmodul
Häufigkeit des Angebotes	jedes Semester
Prüfungsform	Abschlussprüfung
Ermittlung der Modulnote	Benotung der Abschlussprüfung durch die Prüfungskommission
Anerkannte Module	keine
Inhalte	<u>Bachelor-Arbeit</u> Theoretische und/oder experimentelle Arbeit zur Lösung praxisnaher Problemstellungen <u>Mündliche Abschlussprüfung</u> Verteidigung der Bachelor-Arbeit und ihrer Ergebnisse in kritischer Diskussion; Präsentationstechniken
Literatur	fachspezifisch
Weitere Hinweise	<u>Bachelor-Arbeit</u> Dauer der Bearbeitung: 3 Monate gemäß § 29 (8) RSPO <u>Abschlussprüfung</u> Nach Vereinbarung zwischen Prüfling und Prüfungskommission kann die Abschlussprüfung auch auf Englisch erfolgen.

<b>Modulnummer</b>	<b>WP01</b>
<b>Titel</b>	<b>Interaktions- und Interface Design / Interaction and Interface Design</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	<p>Die Studierenden können komplexe Probleme des Interface Designs lösen und Informationen verständlich, rasch erfassbar und gut auffindbar in visuellen Systemen zusammenzufassen. Sie entwickeln bereits erworbene, tiefgreifende Kenntnisse zur plattformübergreifenden Konzeption und Gestaltung von Interface Design für interaktive Anwendungen weiter. Sie sind in der Lage selbständig Interaction Design Techniken und Methoden anzuwenden sowie Designkonzepte für verschiedene Arten interaktiver Medien zu entwickeln. Diese werden den unterschiedlichen, medienpezifischen Anforderungen gerecht und in interaktiven Prototypen umgesetzt.</p> <p>Es werden Kompetenzen zu den Themen Interaction Design, Informationsarchitektur, Usability, User Experience, Interaktivität und Prototyping erworben und die elementaren Faktoren medialer Gestaltung – Objekt, Raum, Zeit und Interaktion – erarbeitet. In einem themenorientierten Projekt werden wichtige Methoden, Aspekte und Phasen zur Gestaltung digitaler Medienanwendungen erlernt und ein interaktiver Prototyp wird erstellt.</p>
<b>Voraussetzungen</b>	Empfehlung: Grundkenntnisse der visuellen Kommunikation und Gestaltung
<b>Niveaustufe</b>	4./5. Studienplansemester
<b>Lehrform</b>	Übung
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 100% gestalterische Übungsaufgaben (14-tägige Übungsblätter).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<p>Interface und Interaktions-Design für unterschiedliche digitale Medienarten, interaktive Medienapplikationen und interaktive Mediensysteme:</p> <ul style="list-style-type: none"> <li>• Informationsarchitektur (Struktur, Navigation, Benutzerführung)</li> <li>• Strukturierung / Komposition (Wireframes, Flowcharts, Storyboards)</li> <li>• Informationsvisualisierung (Methoden zur Planung, Strukturierung und Visualisierung von komplexen Daten und Prozessen)</li> <li>• Kompositionsregeln, medienadäquate Gestaltung, User Experience</li> <li>• Digitaler Content (Gestaltung und Aufbereitung statischer, dynamischer und interaktiver Inhalte)</li> <li>• Animation und Bewegtbild (vor allem im Spannungsfeld von Raum, Zeit, Dynamik, Dramaturgie, Timing, usw.)</li> <li>• Interaktionsdesign (Methoden, linear vs. nonlinear, Orientierung, nutzerorientiertes Design, Ergonomie und Usability,)</li> <li>• Konzeption und Entwurf (grafische Grundbausteine des Interface Designs, Abstraktion &amp; Visualisierung, Interface Design Patterns, Evaluation)</li> <li>• Prototyping und Usability-Testing</li> <li>• Interface Design im interdisziplinären Kontext</li> </ul> <p>In der Übung entwickeln die Studierenden prototypisch eine interaktive Anwendung, die ein komplexes Thema mit visuellen Mitteln und insbesondere unter Einbeziehung von Nutzerinteraktion zeitgemäß und benutzerfreundlich vermittelt.</p>

Literatur	C. Moser: <i>User Experience Design: Mit erlebniszentrierter Softwareentwicklung zu Produkten, die begeistern</i> , Springer Verlag; X.media.press Sendpoints: <i>G.U.I Design: Graphical User Interface Design</i> , Gingko Press Gmbh S. Preuss, T. Leonhardt: <i>THE DIGITAL INNOVATION MODEL - Software planen, die Nutzer lieben</i> , HANDSPIEL Verlag GmbH
Weitere Hinweise	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>WP02</b>
<b>Titel</b>	<b>Software Engineering: Architekturen und Werkzeuge / Software Engineering: Architectures and Tools</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden sollen in der Lage sein, auf Grund von Anforderungen an die zukünftigen Softwaresysteme, komplexe Software-Architekturen zu entwerfen und die Lösungskonzepte für deren statischen und dynamischen Aspekte aufzuzeigen. Dabei beherrschen sie eine Toollandschaft, die für das Aufsetzen von mittelgroßen bzw. großen Projekten notwendig ist und kennen die Abhängigkeiten und die Erstellungsprozesse der im Projekt zu realisierenden Artefakte.
<b>Voraussetzungen</b>	Empfehlung: Software Engineering II
<b>Niveaustufe</b>	4./5. Studienplansemester
<b>Lehrform</b>	Übung
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 100% Klausur (90 Minuten).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<p>Die Modul Inhalte können in folgende Kategorien unterteilt werden:</p> <ul style="list-style-type: none"> <li>• Grundlegende Begriffe</li> <li>• Architekturentwicklung</li> <li>• Architekturmodellierung</li> <li>• Dokumentation der Architekturen</li> <li>• Qualitätskriterien</li> <li>• Typische Softwarearchitekturen</li> <li>• Tools</li> </ul> <p>Nach der Definition der grundlegenden Begriffe wird das grundlegende Vorgehensmodell für die Architekturerstellung vorgestellt. Den Kern der Architekturentwicklung stellt die Erstellung des Architekturentwurfs dar. Hierzu werden die Architektur-Entwurfsmethoden (z. B. Domain-Driven/WAM/Quality-Driven) die wichtigsten Architekturstile und -muster (Datenfluss-Stil/Stile der verteilten Systeme/Ereignisbasierte Systeme usw.) sowie die zu modellierenden Architektur Aspekte und deren Dokumentation ausführlich erläutert. Es werden ebenfalls die Auswahlkriterien für die zu verwendenden Tools bzw. konkrete Toolbeispiele behandelt. Nach der ausführlichen Erläuterung des Erstellungsprozesses für die Softwarearchitekturen werden die Qualitätskriterien sowie die in der Praxis am meisten verwendeten Softwarearchitekturen vorgestellt bzw. diskutiert.</p>
<b>Literatur</b>	G. Starke: <i>Effektive Softwarearchitekturen: Ein praktischer Leitfaden</i> , Carl Hanser Verlag
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>WP03</b>
<b>Titel</b>	<b>3D Web Graphics / 3D Web Graphics</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden erlernen aktuelle Techniken der 3D-Grafikprogrammierung für hardwarebeschleunigte grafisch-interaktive Webanwendungen. Sie lernen die Umsetzbarkeit verschiedener Beleuchtungs- und Materialeffekte mittels aktueller Browser-Schnittstellen einzuschätzen und selbst effiziente Lösungen zu implementieren.
<b>Voraussetzungen</b>	Empfehlung: Programmierung I+II, Mathematik I+II, Computergrafik Grundlagen, Multimedia Engineering I
<b>Niveaustufe</b>	4./5. Studienplansemester
<b>Lehrform</b>	Übung
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 100% schriftliche Übungen (14-tägige Übungsblätter).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<p>Die Schwerpunkte des Moduls liegen auf einem Überblick der 3D-Rasterisierungspipeline sowie auf den wichtigsten Eigenschaften moderner Grafik-Hardware (GPU) und aktueller Low- und High-Level Grafik-Programmierschnittstellen für den Browser, wie z.B. WebGL oder Three.js. Der seminaristische Unterricht vermittelt und vertieft dabei neben der Verwendung besagter Schnittstellen auch die Grundlagen echtzeitfähiger Grafikprogrammierung:</p> <ul style="list-style-type: none"> <li>• Funktionsweise von SIMD und der Rasterisierungs-Pipeline</li> <li>• Geometrische Modellierung mittels Dreiecksnetzen</li> <li>• Transformation, Projektion und Szenengraphen</li> <li>• Beleuchtung und programmierbare Shadereinheiten</li> <li>• Texturierungs- und weitere Effekte</li> </ul> <p>In der Übung wird anhand ausgewählter Beispiele die konkrete Umsetzung der vermittelten Techniken in eigene 3D-Web-Applikationen realisiert.</p>
<b>Literatur</b>	K. Matsuda, R. Lea: WebGL Programming Guide: Interactive 3D Graphics Programming with WebGL (OpenGL), Addison Wesley Weitere aktuelle Literatur wird in der Vorlesung angegeben.
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>WP04</b>
<b>Titel</b>	<b>Softwarequalität und -test / Software Quality and Test</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Heutzutage muss immer mehr Software in immer kürzerer Zeit entwickelt werden. Damit diese Software noch wartbar bleibt, muss von vornherein Augenmerk auf deren Qualität und Testbarkeit gelegt werden. Die Studierenden lernen zu diesem Zweck konstruktive und analytische Methoden der Qualitätssicherung und dafür passende Werkzeuge kennen. Sie können derartige Werkzeuge einordnen, auswählen und anwenden.
<b>Voraussetzungen</b>	Empfehlung: Programmierung I und II, Software Engineering II
<b>Niveaustufe</b>	4./5. Studienplansemester
<b>Lehrform</b>	Übung
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 50% Präsentation (20 Minuten), 50% schriftliche Übungen (14-tägige Übungsblätter).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<p>Software-Qualitätskriterien nach ISO 9126 als Rahmen für die folgenden Inhalte</p> <p>Konstruktive Qualitätssicherung</p> <ul style="list-style-type: none"> <li>• Typsichere Sprachen einsetzen</li> <li>• Programmierkonventionen einhalten (z.B. Google Java Style)</li> <li>• Redundanzfreiheit anstreben (DRY-Prinzip)</li> <li>• Variable Daten minimieren (z.B. mit final, immutable types)</li> <li>• Schnittstellen minimieren</li> <li>• Patterns einsetzen</li> <li>• Test Driven Design</li> </ul> <p>Analytische Qualitätssicherung</p> <ul style="list-style-type: none"> <li>• Code Reviews</li> <li>• Pair Programming</li> <li>• Software-Metriken (z.B. LoC, McCabe Complexity, Fan In, Fan Out, Code Duplications)</li> <li>• Statische Analysewerkzeuge (z.B. SonarQube, PMD/CPD, FindBugs)</li> <li>• Automatisierte Unit-Tests (z.B. mit JUnit oder ScalaTest)</li> <li>• Testüberdeckungsanalyse (z.B. mit SonarQube, Cobertura)</li> <li>• Tests asynchroner Abläufe</li> <li>• Automatisierte Oberflächentests (z.B. mit Selenium)</li> <li>• Automatisierte Lasttests</li> </ul>
<b>Literatur</b>	K. Schneider: <i>Abenteuer Softwarequalität</i> , dpunkt.verlag J. Link: <i>Softwaretests mit JUnit</i> , dpunkt.verlag
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>WP05</b>
<b>Titel</b>	<b>Betriebswirtschaftslehre / Business Administration</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	<p>Die Studierenden kennen den Ablauf einer Unternehmensgründung und die wesentlichen Finanzierungsmöglichkeiten. Sie können situationsgerecht geeignete Rechtsformen auswählen.</p> <p>Sie kennen die Anforderungen an die Organisation und Führung eines Betriebes und können diese auf ihre betrieblichen Praxiserfahrungen übertragen.</p> <p>Die Studierenden kennen die Teilbereiche des betrieblichen Rechnungswesens und deren Aufgaben. Sie sind in der Lage einen (einfachen) Jahresabschluss anhand geeigneter Kennzahlen zu analysieren.</p>
<b>Voraussetzungen</b>	
<b>Niveaustufe</b>	4./5. Studienplansemester
<b>Lehrform</b>	Übung
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 100% Klausur (90 Minuten).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<ol style="list-style-type: none"> <li>1. Unternehmensgründung und Rechtsformwahl</li> <li>2. Unternehmensfinanzierung <ul style="list-style-type: none"> <li>• Überblick über wichtige Finanzierungsarten</li> <li>• Kreditsicherheiten</li> </ul> </li> <li>3. Betriebsorganisation - Grundlagen <ul style="list-style-type: none"> <li>• Betriebliche Aufbauorganisation</li> <li>• Betriebliche Ablauforganisation</li> </ul> </li> <li>4. Personalführung <ul style="list-style-type: none"> <li>• Führungsstile und Mitarbeitergespräche</li> <li>• Zusammenarbeit mit freien Mitarbeitern (Freelancer)</li> </ul> </li> <li>5. Förderung von Innovationen</li> <li>6. Rechnungswesen <ul style="list-style-type: none"> <li>• Grundlagen des betrieblichen Rechnungswesens</li> <li>• Kennzahlen zur Analyse des Jahresabschlusses</li> </ul> </li> </ol>
<b>Literatur</b>	D. Vahs, J. Schäfer-Kunz: <i>Einführung in die Betriebswirtschaftslehre: Lehrbuch mit Beispielen und Kontrollfragen</i> , Schäffer-Poeschel
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>WP06</b>
<b>Titel</b>	<b>Medienproduktion und -distribution / Media Production and Distribution</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden kennen <ul style="list-style-type: none"> <li>• Produktionsprozesse und deren Signalverarbeitung,</li> <li>• die Verteilung der Medien in Systemen und in den Distributionskanälen.</li> </ul>
<b>Voraussetzungen</b>	Empfehlung: Medientechnologien
<b>Niveaustufe</b>	4./5. Studienplansemester
<b>Lehrform</b>	Übung
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 65% Projektarbeit (40 Stunden), 35% schriftliche Übungen (14-tägige Übungsblätter).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<ul style="list-style-type: none"> <li>• Vertiefung der Analoge Audio, Bild und Videotechnik: Signale, Waveform, Wandler (Akustik-Elektrisch), AD/DA-Wandler</li> <li>• Produktionstechnik Kamera/Mikro, Display/Lautsprecher, Mischer, Aufzeichnung und Monitoren, Speicherformat, Übertragung</li> <li>• Produktionsprozesse: Planung und Konzeption, Produktionsdurchführung</li> <li>• Postproduktion: Schnitt, Bearbeitung, Export</li> <li>• Distributionsprozesse: Planung und Konzeption, Produktionsdurchführung</li> <li>• Zwischenprodukte; Metadaten, Archivformat, Kodierungsformat.:</li> <li>• Medien-Systeme: Redaktionssysteme, Asset-Content-Management-Systeme,</li> <li>• Medienprodukte/Dienstleistung</li> </ul>
<b>Literatur</b>	<p>U. Schmidt: Professionelle Videotechnik, Springer Berlin Heidelberg</p> <p>T. Petrasch, J. Zinke: Videofilmproduktion, Carl Hanser Verlag</p> <p>M. Dickreiter: Handbuch der Tonstudiotechnik, De Gruyter SAUR-Verlag</p> <p>H. Henle: Das Tonstudio Handbuch, GC Carstensen Verlag</p> <p>D. Stotz: Computergestützte Audio- und Videotechnik; Multimedialechnik in der Anwendung, Springer Verlag</p>
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.



<b>Modulnummer</b>	<b>WP07</b>
<b>Titel</b>	<b>Effiziente Software Entwickeln mit C++ / Developing Efficient Software with C++</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden lernen, die Programmiersprache C++ als Werkzeug zur Implementierung performanter und modular aufgebauter Applikationen und Bibliotheken zu verwenden. Ein großer Fokus liegt auf aktuellen Sprachstandards und Paradigmen, generischem Programmieren, funktionalem Design, und der Separation of Concerns. Durch die tiefer gehende Betrachtung von Speicherverwaltung, Referenzen und Zeigern, Containern und Polymorphie wird das Grundverständnis für den Performance-bewussten Einsatz der Sprache sowie das für die Funktionsweise anderer Sprachen vertieft.
<b>Voraussetzungen</b>	Programmier-Erfahrung mit Java, JavaScript, o.ä.
<b>Niveaustufe</b>	4./5. Studienplansemester
<b>Lehrform</b>	Übung
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 100% schriftliche Übungen (14-tägige Übungsblätter).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<p>Unterricht und Übungen orientieren sich an folgenden möglichen Themen:</p> <ul style="list-style-type: none"> <li>• Kompilation, Linking, Build-System, Entwicklungsumgebung</li> <li>• Code-Organisation, Präprozessor, Headerdateien</li> <li>• Value-Semantik vs. Referenz-Semantik</li> <li>• Funktionen, Lambdas, funktionales Denken</li> <li>• Klassen und Operatorenüberladung</li> <li>• Modularer Code und Separation of Concerns</li> <li>• Stack, Heap, Zeiger und Referenzen</li> <li>• Ressourcen-Verwaltung, Smart Pointer, RAII</li> <li>• Generisches (Meta-) Programmieren mit Templates</li> <li>• Besonderheiten bei Vererbung; multiple Vererbung</li> <li>• Nutzung von Standard-Bibliotheken für Container, Strings, Algorithmen (STL, Boost)</li> <li>• GUI-Programmierung (Qt/QML)</li> <li>• Eigene Implementierung ausgewählter Algorithmen und Datenstrukturen in C++</li> </ul>
<b>Literatur</b>	B. Stroustrup: <i>The C++ Programming Language</i> , Addison-Wesley. Weitere aktuelle Literatur wird in der Vorlesung vorgestellt.
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten, Englischkenntnisse sind von Vorteil.

<b>Modulnummer</b>	<b>WP08</b>
<b>Titel</b>	<b>Programmiersprachen und -paradigmen / Programming Languages and Paradigms</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Das Ziel dieser Veranstaltung ist die Vermittlung der grundlegenden Programmierparadigmen und deren Besonderheiten. Die Studierenden sollen in der Lage sein, eine beliebige, für sie unbekannte Programmiersprache, schnell zu überblicken, deren wichtigsten Merkmale zu erkennen und in dieser Sprache innerhalb kurzer Zeit (max. 2 Wochen) Programme zu entwickeln. Dabei können sie mindestens 80% der Sprachkonstrukte der neuen Sprache aktiv verwenden.
<b>Voraussetzungen</b>	Empfehlung: Programmieren I/II
<b>Niveaustufe</b>	4./5. Studienplansemester
<b>Lehrform</b>	Übung
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 100% Klausur (90 Minuten).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<p>Das Modul ist wie folgt strukturiert:</p> <ol style="list-style-type: none"> <li>1.) Geschichte der Programmiersprachen, Grundbegriffe: Syntax, Semantik, Typen, Typsysteme usw.</li> <li>2.) Implementierungsgrundlagen von Programmiersprachen, Compiler vs. Interpreter</li> <li>3.) Programmierparadigmen und ihre Besonderheiten</li> <li>4.) Imperatives Paradigma</li> <li>5.) Deklaratives Paradigma</li> <li>6.) Funktionales Programmierparadigma</li> <li>7.) Relationales/Logisches Programmierparadigma</li> <li>8.) Objektorientiertes Paradigma</li> </ol> <p>Zu jedem Paradigma werden die wichtigsten Sprachkonzepte und deren konkreten Beispiele in den ausgewählten Programmiersprachen vorgestellt. Die konkreten Sprachen und die zu behandelnden Konzepte wählt dabei der Dozent aus. Beispiel für das funktionale Programmierparadigma:</p> <ul style="list-style-type: none"> <li>• Sprachkonzepte: Referentielle Transparenz, Lambdakalkül als Kern jeder funktionalen Sprache, Funktionen als Objekte erster Ordnung, Curried Functions, Evaluationsstrategien.</li> <li>• Programmiersprachen: ML, Haskell.</li> </ul>
<b>Literatur</b>	<p>A. Clausen: <i>Programmiersprachen; Konzepte, Strukturen und Implementierung in Java</i>, Spektrum Akademischer Verlag</p> <p>B. A. Tate: <i>Seven Languages in Seven Weeks: A Pragmatic Guide to Learning Programming Languages</i>, The Pragmatic Programmers</p>
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>WP09</b>
<b>Titel</b>	<b>Anwendungsentwicklung für IOS-Geräte / Application Development for IOS Devices</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden sollen in der Lage sein, iPhone- und iPad-Applikationen aufzusetzen und datenbank-, ortungs-, und kartebasierte Apps ohne Einarbeitungsaufwand zu entwickeln. Sie sind in der Lage, unbekannte iOS-Programmibliotheken (wie z. B. Bibliotheken für die Erstellung von WEB-Applikationen) innerhalb einer Woche produktiv zu verwenden.
<b>Voraussetzungen</b>	
<b>Niveaustufe</b>	4./5. Studienplansemester
<b>Lehrform</b>	Übung
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 100% Klausur (90 Minuten).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<p>Im Rahmen der Veranstaltung werden folgende Themen behandelt:</p> <ul style="list-style-type: none"> <li>• Grundlegenden Prinzipien der Programmiersprache Swift mit den entsprechenden Sprachkonstrukten</li> <li>• Wichtigste Softwarebibliotheken für die iOS-Geräte GUI, GPS, Kamera...</li> <li>• Verwendung des Softwareentwicklungstools xCode.</li> </ul> <p>Jeder Unterrichtsblock beinhaltet eine theoretische Einführung und das entsprechende praktische Anwendungsszenario – eine kleine Beispiel-App. Die Entwicklung der Beispiel-App wird durch den Dozenten vorgeführt und danach durch die Studierenden an eigenen Rechnern wiederholt.</p> <p>Anschließend wird eine App entwickelt, welche die gezeigten Konzepte und Bibliotheken in ein Projekt integriert.</p>
<b>Literatur</b>	<i>The Swift Programming Language, Swift 2.2 Edition</i> , Apple Inc., Cupertino, CA K. M. Rodewig, Clemens Wagner: <i>Apps programmieren für iPhone und iPad: Inkl. Xcode, Debugging, Versionierung, zahlreiche Praxisbeispiele. Aktuell zu iOS 8 und inkl. Einführung in Swift</i> , Galileo Computing
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>WP10</b>
<b>Titel</b>	<b>Mobile Anwendungsentwicklung / Mobile Application Development</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	<ul style="list-style-type: none"> <li>• Die Studierenden kennen typische Anwendungsszenarien für mobile Anwendungen und können Mehrwerte mobiler Anwendungen im Hinblick auf diese Szenarien identifizieren.</li> <li>• Die Studierenden verfügen über Entwicklungskompetenzen in einer ausgewählten Technologie zur Umsetzung mobiler Applikationen (z.B. Android Java Framework, Webtechnologien und -frameworks) und können Anwendungen mit Standard-Bedienelementen wie Listen, Formularen, Menüs, Dialogen etc. entwickeln.</li> <li>• Die Studierenden verstehen die Rolle mobiler Applikationen im Rahmen von Client-Server Architekturen und können Alternativen zur client- und server-seitigen Persistierung der von einer Anwendung verwendeten Daten umsetzen.</li> <li>• Die Studierenden kennen – ausgehend vom Beispiel der verwendeten Technologie – Einsatzmöglichkeiten und Architekturmuster von Anwendungsframeworks.</li> <li>• Die Studierenden sind dazu in der Lage, die in der Veranstaltung erworbenen Entwicklungskompetenzen im Bereich der ausgewählten Technologie selbständig anhand von einschlägigem Dokumentationsmaterial weiterzuentwickeln.</li> </ul>
<b>Voraussetzungen</b>	Empfehlung: Programmierung I+II, Verteilte Systeme, Datenbanksysteme, Web Engineering I
<b>Niveaustufe</b>	4./5. Studienplansemester
<b>Lehrform</b>	Übung
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 30% Klausur (90 Minuten), 70% schriftliche Übungen (14-tägige Übungsblätter).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<p>Behandelt wird eine Auswahl der für den Einsatz und die Entwicklung mobiler Anwendungen relevanten Aspekte, z.B.:</p> <ul style="list-style-type: none"> <li>• Mobile Datenübertragung</li> <li>• Mobile Endgeräte</li> <li>• Anwendungsszenarien mobiler Applikationen</li> <li>• Sicherheitsaspekte und Nutzung privater Daten in mobilen Anwendungen</li> <li>• Grundlagen einer ausgewählten Technologie zur Entwicklung mobiler Applikationen</li> <li>• Listenansichten</li> <li>• Formulare und Formularvalidierung</li> <li>• Menüs und Dialoge</li> <li>• Lokale und server-seitige Datenspeicherung</li> <li>• Kartenansichten</li> <li>• Architektur von Frameworks</li> </ul> <p>Als Technologie zur Umsetzung von Anwendungen werden entweder das Android Java Framework oder Webtechnologien und -frameworks verwendet.</p>

Literatur	<p>Android Entwickler-Website. <a href="http://developer.android.com/develop/index.html">http://developer.android.com/develop/index.html</a></p> <p>J. Annuzi, L. Darcey, S. Conder: <i>Introduction to Android Application Development: Android Essentials (Developer's Library)</i>. Addison Wesley</p> <p>T. Bollmann, K. Zeppenfeld: <i>Mobile Computing – Hardware, Software, Kommunikation, Sicherheit, Programmierung</i>. W3L Verlag</p> <p>P. Gaston: <i>The Modern Web – Multi-Device Web Development with HTML5, CSS3, and JavaScript</i>. O'Reilly</p> <p>Mozilla Developer Network. <a href="https://developer.mozilla.org">https://developer.mozilla.org</a></p> <p>R. Oechsle: <i>Java Komponenten – Grundlagen, prototypische Realisierung und Beispiele für Komponentensysteme</i>. Hanser</p> <p>O. Zeigermann: <i>JavaScript für Java-Entwickler</i>. <a href="http://entwickler.press">entwickler.press</a></p>
Weitere Hinweise	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>WP11</b>
<b>Titel</b>	<b>Visual and Scientific Computing / Visual and Scientific Computing</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden erlernen anhand ausgewählter Fragestellungen Grundlagen des Visual Computing und des wissenschaftlichen Rechnens.
<b>Voraussetzungen</b>	Empfehlung: Ein erfolgreicher Abschluss der Module Mathematik 1 + 2, Programmieren 1 + 2 und Computergrafik Grundlagen
<b>Niveaustufe</b>	4./5. Studienplansemester
<b>Lehrform</b>	Übung
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 100% schriftliche Übungen (14-tägige Übungsblätter).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	In dem Modul werden angewandte Probleme aus wechselnden Themenbereichen der Computergrafik, Bildverarbeitung, Visualisierung, der virtuellen und erweiterten Realität oder des maschinellen Lernens untersucht. Dabei wird das Verständnis für aktuelle wissenschaftliche Texte gefördert sowie einige Grundlagen wissenschaftlichen Rechnens wie z.B. Ausgleichsrechnung, Eigenwertprobleme oder Optimierungsverfahren vermittelt und durch Programmierübungen verfestigt.
<b>Literatur</b>	Literatur wird während der Veranstaltung bekannt gegeben.
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>WP12</b>
<b>Titel</b>	<b>Spieleentwicklung und Creative Coding / Game Development and Creative Coding</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden lernen die grundlegenden Architektur- und Entwurfsmuster von aktuellen Rahmenwerken und Bibliotheken aus den Bereichen Creative Coding, Spieleentwicklung und Simulation kennen. Damit sind Sie in der Lage, existierende Systeme zu verwenden und zu erweitern, sowie eigene Projekte unter Verwendung relevanter Best Practices umzusetzen.
<b>Voraussetzungen</b>	Empfehlung: Programmierkenntnisse
<b>Niveaustufe</b>	4./5. Studienplansemester
<b>Lehrform</b>	Übung
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 100% schriftliche Übungen (14-tägige Übungsblätter).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<p>In diesem Modul wird die Architektur aktueller Media- und Game-Engines exemplarisch dargestellt. Dabei wird besonders auf die technischen Grundlagen einzelner Komponenten eingegangen. Mögliche Themenbereiche sind:</p> <ul style="list-style-type: none"> <li>• Architektur- und Entwurfsmuster</li> <li>• Real-Time Rendering</li> <li>• Physikalische Simulation und Animation</li> <li>• Game AI und Networking</li> <li>• Echtzeit Video- und Bildbearbeitung</li> <li>• Tool-Chain und externe Formate</li> <li>• Entwicklung für mobile Geräte</li> </ul> <p>In den Übungen entwickeln die Studierenden semesterbegleitend in kleinen Gruppen das Konzept und den Prototypen einer eigenen interaktiven Multimedia-Anwendung, eines Computerspiels oder einer Simulation unter Einsatz aktueller Rahmenwerke und Bibliotheken.</p>
<b>Literatur</b>	<p>D. H. Eberly, <i>3D Game Engine Architecture</i>, Morgan Kaufmann.  I. Millington, <i>Game Physics Engine Development</i>, Morgan Kaufmann.  R. Madeira, D. Gorny, <i>Cinder Creative Coding Cookbook</i>, Packt Publishing.  J. Gibson, <i>Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C#</i>, Addison Wesley.</p>
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>WP13</b>
<b>Titel</b>	<b>Signalverarbeitung für Audio, Bild und Video / Signal Processing for Audio, Image and Video</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden kennen die Programmierung der Digitalen Signale: <ul style="list-style-type: none"> <li>• Programmierung der digitalen Signalverarbeitung</li> <li>• Programmierung der Kompression</li> </ul>
<b>Voraussetzungen</b>	Empfehlung: Medientechnologien
<b>Niveaustufe</b>	4./5. Studienplansemester
<b>Lehrform</b>	Übung
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 65% Projekt (40 Stunden), 35% schriftliche Übungen (14-tägige Übungsblätter).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<ul style="list-style-type: none"> <li>• Analog, AD-Wandlung, Wandler-Prinzip, Aliasing,</li> <li>• Signalkette, 1D, 2D, 3D, GoP, Vector, Matrix</li> <li>• Audio-API, Bild-API, Video-API: aktuelle Programmieretechnik (u.a. JavaScript)</li> <li>Algorithmen: Digitale Signalverarbeitung</li> <li>• ABV-Bearbeitung: Mischen, Up-/Downsampling, Blende</li> <li>• ABV-Effekte: Filter, Zeit-Effekte, Frequenz-Effekte, Schwellwert: Bluescreen</li> <li>• ABV-Analyse von Signalen: Waveform, Histogramm, FFT/DCT,</li> <li>• einfache Medienobjekterkennung: Phonem, Wort, Satz, Kanten, Bildkonturen, Bewegung,</li> <li>• ABV-Kompression: Prädiktion, DeltaPCM, Subband, Transformation DFT, DCT und MDCT, Bild-Wavletkodierung, Video-Bewegungsschätzung und Kompensation</li> <li>• Qualität, Fehlerberechnung: SAD, PSNR</li> </ul>
<b>Literatur</b>	U. Schmidt: <i>Professionelle Videotechnik</i> , Springer Berlin Heidelberg T. Strutz: <i>Bilddatenkompression. Grundlagen, Codierung, JPEG, MPEG, H264</i> , Vieweg-Verlag S. Pfeiffer: <i>The Definitive Guide to HTML5 for Video (and Audio)</i> , Apress Verlag
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.



<b>Modulnummer</b>	<b>WP14</b>
<b>Titel</b>	<b>Maschinelles Lernen / Machine Learning</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	<p>Die Studierenden können den Begriff des maschinellen Lernens konkretisieren und von dem des menschlichen Lernens abgrenzen. Sie lernen eine Reihe unterschiedlicher Verfahren des maschinellen Lernens kennen und erproben diese an Beispielen in Unterricht und Übung.</p> <p>Sie können die Randbedingungen des maschinellen Lernens erläutern und beurteilen, welche Verfahren am besten für bestimmte Probleme geeignet sind. Sie verstehen was "richtig" lernen im algorithmischen Sinne bedeutet (z.B. nicht auswendig lernen).</p> <p>In den Übungen können kleine Projekten von 1-3 Wochen Länge kleinere lernende Systeme programmiert (oder vorbereitete Systeme fertiggestellt) und an praktischen Beispielen ausprobiert werden. (Beispiele: Erstellen eines Entscheidungsbaums (händisch/programmatisch) zu gegebenen Datensätzen; Implementation/Konfiguration eines neuronalen Netzes zur Erkennung von Buchstaben; Implementation einer Warenkorbanalyse mit Hilfe von Assoziationsregeln, ...). Neben praktischen Übungen werden theoretische Konzepte vertieft.</p> <p>Kenntnisse im Maschinellen Lernen bieten Berufschancen beispielsweise für Anwendungen in Verkaufsportalen wie Zalando oder Suchmaschinen wie Google.</p>
<b>Voraussetzungen</b>	Gute Programmierkenntnisse, algorithmische Vorbildung, mathematische Grundkenntnisse.
<b>Niveaustufe</b>	4./5. Studienplansemester
<b>Lehrform</b>	Übung
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 50% Klausur (90 Minuten), 50% schriftliche Übungen (14-tägige Übungsblätter).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<ul style="list-style-type: none"> <li>• Lernen aus Daten</li> <li>• Visualisierung von Daten</li> <li>• Unsupervised Learning - Clustering</li> <li>• Klassifizierung</li> <li>• Vorhersagen - Approximation von Funktionen</li> <li>• Genetische Algorithmen</li> <li>• Verknüpfen von Datensätzen - Association Rules</li> <li>• Theoretische Betrachtungen</li> </ul>
<b>Literatur</b>	<p>P.-N. Tan, M. Steinbach, V. Kumar: <i>Introduction to Data Mining</i></p> <p>T. M. Mitchell: <i>Machine Learning</i>, McGraw-Hill Science/Engineering/Math</p> <p>S. Marsland: <i>Machine Learning: An Algorithmic Perspective</i>, Crc Pr Inc</p>
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>WP15</b>
<b>Titel</b>	<b>Aktuelle Webtechnologien: Frameworks und Tools / Current Web Technologies: Frameworks and Tools</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden konzipieren und realisieren Webanwendungen mit modernen Werkzeugen und Frameworks. Die Studierenden können komplexe Architektur- und Entwurfsmuster für Webanwendungen wiedergeben, abwägen und anwenden.
<b>Voraussetzungen</b>	Empfehlung: Programmieren I+II, Algorithmen, Software Engineering I+II, Web Engineering I+II
<b>Niveaustufe</b>	5. Studienplansemester
<b>Lehrform</b>	Übung
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 30% Präsentation (20 Minuten), 50% schriftliche Übungen (14-tägige Übungsblätter), 20% Laborbericht (5-8 Seiten).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<p>In dem praktisch ausgerichteten Modul werden die Lehr-/Lerninhalte in wesentlichen Teilen durch die Studierenden erarbeitet. Die wechselnden Themen werden in Kleingruppen durch wissenschaftliche Recherche (u.A. Internet) aufbereitet und im Rahmen von Lehrvorträgen und kleinen Übungsaufgaben untereinander vermittelt. In kleinen Teams werden eigene Prototypen unter Nutzung der aktuellen Frameworks mittels agiler Entwicklungsmethoden wöchentlich um die neu erlernten Inhalte ergänzt. Die Lehrkraft nimmt dabei die Rolle eines „Kunden“ ein, welche die Produktprototypen anhand von Meilensteinen abnimmt.</p> <p>Im jeweiligen Semester wird immer eines (bzw. wenige) Frameworks, -tools betrachtet und mit diesen intensiv gearbeitet. Die aktuelle Auswahl wird zu Beginn der Lehrveranstaltung von der Lehrkraft mitgeteilt.</p> <p>Die praktische Arbeit im Rahmen der wöchentlichen Übungen umfasst:</p> <ul style="list-style-type: none"> <li>• Einarbeitung in Teilaspekte moderner Webframeworks und -tools anhand von Dokumentation und Codebeispielen (unter Anleitung)</li> <li>• Entwurfsprinzipien und Architekturdesigns der Webframeworks und -tools</li> <li>• Recherche und Problemlösestrategien</li> <li>• Kollegiale Beratung in der Projektarbeit</li> <li>• Strategisches Fehlerlösen (Debuggingstrategien)</li> <li>• Agile Konzeption und Entwicklung eigener Prototypen im Team anhand der vorgestellten Teilaspekte moderner Webframeworks und -tools.</li> <li>• Dokumentation und Werkzeuge zur Dokumentation</li> </ul>
<b>Literatur</b>	Die Literatur sowie die zu empfehlenden Internet-Ressourcen werden semesteraktuell durch die Lehrkraft festgelegt.
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>WP16</b>
<b>Titel</b>	<b>Aktuelle Themen / Selection of Current Issues</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden kennen Aktuelle Methoden und Technologien aus den Bereichen Webentwicklung, mobile Anwendungen, Computergrafik, Medientechnologien. Die Studierenden lernen selber Themen zu erarbeiten.
<b>Voraussetzungen</b>	Empfehlung: gute Programmierkenntnisse
<b>Niveaustufe</b>	4./5. Studienplansemester
<b>Lehrform</b>	Übung
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 50% Präsentation (20 Minuten), 50% schriftliche Übungen (14-tägige Übungsblätter).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<ul style="list-style-type: none"> <li>• Kurzpräsentationen zu aktuellen Themen durch Studierende, externe Fachleute, Lehrkräfte</li> <li>• Übungen und Tutorials zu den aktuellen Themen</li> </ul>
<b>Literatur</b>	Wird in der Lehrveranstaltung bekannt gegeben.
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>WP17</b>
<b>Titel</b>	<b>Frontend-Design Web: Fortgeschrittene Techniken / Frontend-Design Web: Advanced Techniques</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden erlernen fortgeschrittene Techniken für das Frontend-Design im Bereich Webprogrammierung. Sie erwerben theoretische und praktische Kenntnisse in Bezug auf die Anwendung von modernen Layout-Techniken und den Umgang mit Präprozessoren. Der sichere Umgang mit den vermittelten Techniken befähigt die Studierenden dazu, wartbare, wiederverwendbare und performante CSSDateien zu erstellen. Durch die intelligente Auswahl von Frameworks und Libraries oder auch den Verzicht darauf bekommen die Studierenden mehr Spielraum für eigenes Design, das nicht durch gängige Frameworks wie bootstrap oder foundation vorgegeben ist.
<b>Voraussetzungen</b>	
<b>Niveaustufe</b>	4./5. Studienplansemester
<b>Lehrform</b>	Übung
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 100% Projektarbeit (40 Stunden)
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<ul style="list-style-type: none"> <li>• Fortgeschrittene Techniken für das Frontend-Design im Bereich Webprogrammierung.</li> <li>• Anwendung und Vergleich neuester CSS3 Layout-Module, wie flex, grid, shapes und multicolumn.</li> <li>• Verwendung von CSS3 Farb-Modulen anstelle von Grafiken und Hintergründen.</li> <li>• Festigung im Umgang mit strukturellen Pseudo-Klassen und Spezifitätsregeln.</li> <li>• Einführung in den Umgang mit Präprozessoren. Was ist ein Präprozessor, welche gibt es und wie kann man gängige Programmierkonzepte auf CSS3 anwenden, so daß die Dateien effizient und wartbar werden.</li> <li>• Praktische Anwendung eines Präprozessors mit Sass (Syntactically awesome stylesheets), einer Scriptsprache für die Erweiterung von CSS-Stylesheets zu wartbaren, redundanzfreien und performanten Dateien, durch Verwendung von Variablen, Funktionen und Kontrollstrukturen.</li> <li>• Aufsetzen des Sass-Compilers und eines Projektes.</li> </ul> <p>Die Techniken werden an Hand eines praktischen Beispiels vorgestellt, und dann mit einer Übungsaufgabe gefestigt. Die Übungen erfolgen in Einzelarbeit, damit das Erlernete auch verinnerlicht werden kann.</p>
<b>Literatur</b>	Literatur wird während der Veranstaltung bekannt gegeben.
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>WP18</b>
<b>Titel</b>	<b>Microservice-Entwicklung: Architekturen und Entwurfsmuster / Developing Microservices: Architectures and Design Patterns</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden erlernen folgende Kenntnisse und Fertigkeiten: <ul style="list-style-type: none"> <li>• die Begriffe Microservices, SOA und Virtualisierung</li> <li>• die Herausforderungen bei der Entwicklung von Microservices</li> <li>• Microservices Planen und Entwerfen</li> <li>• Standard-Schnittstellen (REST, WebServices, etc.)</li> <li>• Deployment-Methoden Planen und Anwenden</li> <li>• die Container-Umgebungen und Betriebsmodelle für Microservices Planen und Betreiben (Docker, VM-Ware, Kubernetes)</li> <li>• Patterns für den Microservice-Entwurf</li> <li>• die Rolle des Devops</li> </ul>
<b>Voraussetzungen</b>	
<b>Niveaustufe</b>	4./5. Studienplansemester
<b>Lehrform</b>	Übung
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 70% Projektarbeit (40 Stunden), 30% Präsentation (20 Minuten).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	Bei der Bereitstellung und Entwicklung von Services sind Flexibilität und Skalierbarkeit wichtige Kriterien. Durch die Verfügbarkeit von Virtualisierungs-Methoden für die Hardware wie z.B. Docker, wird der Bedarf an Applikationen, die auf die Ressourcen abgestimmt sind, immer größer. Aktuelle Software-Architekturen antworten auf diese Anforderungen durch den Entwicklungsansatz 'Microservices', deren Entwurf, Entwicklung, Deployment und Betrieb Gegenstand dieses Kurses sind. Folgende Themen werden behandelt: <ul style="list-style-type: none"> <li>• Was sind Microservices?</li> <li>• Das Gesetz von Conway</li> <li>• Herausforderungen bei Microservices</li> <li>• Microservices und SOA</li> <li>• Microservice-System-Architekturen</li> <li>• Wichtige Entwurfs-Muster für den Entwurf von Microservices</li> <li>• Kommunikations- und Integrations-Schnittstellen</li> <li>• Testen von Microservices</li> <li>• Devops: Entwickelnde Admins, administrierende Entwickler</li> <li>• Betrieb und Continuous Delivery von Microservices / Docker</li> </ul>
<b>Literatur</b>	B. Wolff: <i>Microservices - Grundlagen flexibler Softwarearchitekturen</i> , dpunkt.verlag Bredan Burns: <i>Verteilte Systeme mit Kubernetes entwerfen</i> , O'Reilly - Dpunkt.verlag
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>WP19</b>
<b>Titel</b>	<b>Interactive Media Objects &amp; Web-Media Applications</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden erlernen aktuelle Technologien der kreativen Konzeption, multimedialer Gestaltung und vor allem Programmierung von sog. audiovisuellen Objekten (Video oder Audio 'Chunks', 3D-Objekte bei Sound, Multi-Layer & Multi-Channel bei Video/Bild/Animation-Objekten usw.) für interaktive Web-Media Anwendungen. Sie lernen die Umsetzbarkeit von web-basierten Verfahren kennen, um audiovisuelle Medien als Streams und Medienobjekte mit vielfältigen Interaktionstechniken zu nutzen und dynamisch audiovisuell zu manipulieren.
<b>Voraussetzungen</b>	Empfehlung: Medientechnologien
<b>Niveaustufe</b>	4. Studienplansemester
<b>Lehrform</b>	Übung
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 65% Projektarbeit (40 Stunden), 35% Präsentation (20 Minuten).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	In der Übung: <ol style="list-style-type: none"> <li>1. Konzeptions- und Entwurfstechniken für interaktive Web-Media Anwendungen</li> <li>2. Contenterstellung und Live-Medienproduktion: Pre-, Post-Produktion, Export, Archiv, Kompression für Distributionswege</li> <li>3. Schwerpunkt 1: Interaktivität von audiovisuellen Elementen bei objektbasierten Medienapplikationen</li> <li>4. Schwerpunkt 2: Echtzeit Content-Veränderungen und Inhalts-Modifikationen von Audiovisuellen Objekten basierend auf den ausgeführten Interaktionen</li> <li>5. Schwerpunkt 3: 3D, Multi-Layer, Multi-Channel Techniken bei audiovisuellen Objekten</li> <li>6. Programmierung von kleinen Modulen für audio-visuelle Applikationen, wie Web-Anwendungen, Streaming-Anwendungen, Mediathek-Portalen, Web-Conferencing, VR-Media, u.a. in Javascript und mit FFMPEG</li> </ol>
<b>Literatur</b>	Die aktuelle Literatur wird zu Beginn des Semesters von der Lehrkraft angegeben.
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.

<b>Modulnummer</b>	<b>WP20</b>
<b>Titel</b>	<b>Webprogrammierung mit Python / Web Programming with Python</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden lernen Web-Applikationen mit der Programmiersprache Python zu entwickeln. Sie können verschiedene aktuelle Technologien bzw. Werkzeuge wie Entwicklungs-Frameworks, Verknüpfung zur Datenbank, ORM (object-relational mapping) und Maßnahmen zur Sicherheit einsetzen. Sie können den Inhalt einer Web-Applikation nach dem Entwurfsmuster „Model-View-Controller“ aufbauen und mit interaktiven und multimedialen Elementen gestalten. Sie können eine Web-Applikation von der Konzeption bis zur Umsetzung im Team realisieren. Außerdem wird die Sozialkompetenz der Studierenden durch Gruppenarbeit gefördert.
<b>Voraussetzungen</b>	
<b>Niveaustufe</b>	4./5. Studienplansemester
<b>Lehrform</b>	Übung
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 70% schriftliche Übungen (14-tägige Übungsblätter), 30% Klausur (90 Minuten).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Anerkannte Module</b>	Module vergleichbaren Inhalts
<b>Inhalte</b>	<ul style="list-style-type: none"> <li>• Grundlagen der Programmiersprache Python (Syntax, Operatoren, Kontrollstrukturen, Datenstrukturen, Ausnahmebehandlung)</li> <li>• Python-basiertes Entwicklungs-Framework (z.B. Django)</li> <li>• Gestaltung des Inhalts einer Web-Applikation nach dem Entwurfsmuster „Model-View-Controller“</li> <li>• Interaktion mit Benutzern (z.B. Formular, E-Mail, Datei hoch- und herunterladen)</li> <li>• Multimedialer Inhalt (z.B. visuelle Effekte, Klangeffekte, Musik, Video)</li> <li>• Personalisierung mittels Cookies und Sessions, dazu Maßnahmen zur Sicherheit</li> <li>• Verknüpfung zur Datenbank mit ORM (Object-Relational Mapping)</li> <li>• Qualitätssicherung entsprechend ausgewählter Entwicklungs-Frameworks</li> <li>• Optional: Weitere vertiefende Themen (z.B. Datenbehandlung mit XML / JSON; Einsatz einer REST-API; Suchfunktion; Standard-Bibliothek mit dem Schwerpunkt mathematische Berechnungen und Leistungsfähigkeit; virtuelle Umgebung)</li> </ul> <p>In den Übungen werden die Themen am Rechner vertieft. Die Studierenden entwickeln in kleinen Gruppen eine Web-Applikation.</p>
<b>Literatur</b>	Paul Barry: <i>Python von Kopf bis Fuß</i> . O'Reilly. Nigel George: <i>Build Your First Website with Django 2.1</i> . GNW Independent Publishing. Nigel George: <i>Mastering Django 2: Core</i> . (In Bearbeitung.) GNW Independent Publishing. David Beazley, Brian K. Jones: <i>Python Cookbook</i> . O'Reilly.
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.
<b>Raumbedarf</b>	

<b>Modulnummer</b>	<b>WP21</b>
<b>Titel</b>	<b>Echtzeitgrafik Grundlagen / Fundamentals of Real-Time Rendering</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
<b>Verwendbarkeit</b>	
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden erlernen aktuelle Techniken der 3D-Grafikprogrammierung für hardware-beschleunigte grafisch-interaktive Webanwendungen. Sie lernen die Umsetzbarkeit verschiedener Beleuchtungs- und Materialeffekte mittels aktueller Browser-Schnittstellen einzuschätzen und selbst effiziente Lösungen zu implementieren.
<b>Voraussetzungen</b>	Empfehlung: Programmierung I+II, Mathematik I+II, Computergrafik Grundlagen, Multimedia Engineering I
<b>Niveaustufe</b>	4./5. Studienplansemester
<b>Lehr- und Lernform</b>	Übung
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 100% schriftliche Übungen (14-tägige Übungsblätter).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Inhalte</b>	<p>Die Schwerpunkte des Moduls liegen auf einem Überblick der 3D-Rasterisierungs-Pipeline sowie auf den wichtigsten Eigenschaften moderner Grafik-Hardware (GPU) und aktueller Low- und High-Level Grafik-Programmierschnittstellen für den Browser, wie z.B. WebGL oder Three.js. Der seminaristische Unterricht vermittelt und vertieft dabei neben der Verwendung besagter Schnittstellen auch die Grundlagen echtzeitfähiger Grafikprogrammierung:</p> <ul style="list-style-type: none"> <li>• Funktionsweise von SIMD und der Rasterisierungs-Pipeline</li> <li>• Geometrische Modellierung mittels Dreiecksnetzen</li> <li>• Transformation, Projektion und Szenengraphen</li> <li>• Beleuchtung und programmierbare Shadereinheiten</li> <li>• Texturierungs- und weitere Effekte</li> </ul> <p>In der Übung wird anhand ausgewählter Beispiele die konkrete Umsetzung der vermittelten Techniken in eigene 3D-Web-Applikationen realisiert.</p>
<b>Literatur</b>	K. Matsuda, R. Lea: WebGL Programming Guide: Interactive 3D Graphics Programming with WebGL (OpenGL), Addison Wesley Weitere aktuelle Literatur wird in der Vorlesung angegeben.
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.
<b>Raumbedarf</b>	



<b>Modulnummer</b>	<b>WP22</b>
<b>Titel</b>	<b>Professionelle Web-Anwendungen umsetzen und betreiben / Implementing and maintaining professional web applications</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
<b>Verwendbarkeit</b>	
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Lernziele / Kompetenzen</b>	<p>Das Wahlpflichtfach setzt unmittelbar auf dem Pflichtmodul Web-Engineering II auf, wo die Studierenden lernen eine Single-Page-Anwendung und einen REST-Server umzusetzen.</p> <p>Für den Einsatz einer solchen Anwendung im professionellen Umfeld gibt es zahlreiche Aspekte, die beachtet und umgesetzt werden müssen. In diesem Modul lernen die Studierenden, welche Aspekte für eine professionelle Weiterentwicklung und den Betrieb einer Web-Anwendung von Bedeutung sind. Anhand einer beispielhaften Web-Anwendungen werden Techniken und Methoden wie beispielsweise automatisiertes Testen, Continuous Integration, Hardening und horizontales Skalieren praktisch umgesetzt.</p>
<b>Voraussetzungen</b>	Empfehlung: Web-Engineering I + II
<b>Niveaustufe</b>	4./5. Studienplansemester
<b>Lehr- und Lernform</b>	Übung
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 100% schriftliche Übungen (14-tägige Übungsblätter).
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Inhalte</b>	<p>Es werden unter anderem die folgenden Themen behandelt</p> <ul style="list-style-type: none"> <li>• IT-Security und Hardening von Web-Anwendungen: Schützen der Anwendung und des REST-Servers vor gängigen Hacker-Angriffen</li> <li>• Skalierung: Deployment der Anwendung in einem horizontalen Cluster (z.B. mit Docker und Kubernetes)</li> <li>• Automatisiertes Testen, Continuous Integration/ Deployment: Unit-, Integrations-, Performance- und End-2-End-Tests, für Web-Anwendung und REST-Server über gängige Testing-Tools, CI-Pipelines in Git, etc.</li> <li>• UX – User Experience: Grundregeln einer professionellen User-Experience</li> <li>• Monitoring für Web-Anwendungen: Nutzen von gängigen Stack-Komponenten für das Logging, Performance-Monitoring und Nutzer-Monitoring</li> <li>• Weiterführende Themen: Übersichtsartig könnten noch folgende Themen behandelt werden: Micro-Services, Micro-Frontends, Performance-Monitoring, High-Performance-Logging-Server ...</li> </ul> <p>In den Übungsaufgaben wird eine Beispielanwendung umgesetzt. Für jede Übungsaufgabe werden ein kurzer Bericht sowie die erstellten Artefakte eingereicht und benotet.</p>
<b>Literatur</b>	Weitere aktuelle Literatur wird in der Vorlesung angegeben.
<b>Weitere Hinweise</b>	Dieses Modul wird auf Deutsch angeboten.
<b>Raumbedarf</b>	

<b>Modulnummer</b>	<b>WP23</b>
<b>Titel</b>	<b>Data Science Programmierung mit Python / Data Science Programming with Python</b>
<b>Leistungspunkte</b>	5 LP
<b>Workload</b>	68 Stunden Präsenz (4 SWS Ü), 82 Stunden Selbststudium
<b>Verwendbarkeit</b>	Eigener Studiengang
<b>Lerngebiet</b>	Fachspezifische Grundlagen
<b>Qualifikationsziele / Kompetenzen</b>	Die Studierenden lernen Datenverarbeitung mit der Programmiersprache Python zu realisieren. Wichtig sind dabei die Konzepte, die eine Arbeit als Data Scientist ermöglichen. Sie können verschiedene aktuelle Bibliotheken verwenden. Sie können ihre Programmierlösung in Jupyter Notebook, der Standard-Entwicklungsumgebung zu Data Science, gestalten und umsetzen. Sie können Daten nach Aufgabenstellung durch Analyse, Bearbeitung und Umgestaltung bereitstellen sowie ihre Einsichten durch Visualisierung darstellen.
<b>Voraussetzungen</b>	Empfehlung: Programmierung I + II
<b>Niveaustufe (Dauer)</b>	4./5. Studienplansemester (einsemestrig)
<b>Lehr- und Lernform</b>	Übung
<b>Status</b>	Wahlpflichtmodul
<b>Häufigkeit des Angebotes</b>	jedes Semester
<b>Prüfungsform</b>	Die Prüfungsform wird nach §19(2) RSPO durch die Lehrkraft festgelegt. Sofern die Lehrkraft die Prüfungsform und die Prüfungsmodalitäten nicht am Semesteranfang in der Frist nach §19 (2) RSPO festlegt, gilt folgende Prüfungsform: 100% schriftliche Übungsaufgaben.
<b>Ermittlung der Modulnote</b>	Siehe Studienplan
<b>Inhalte</b>	<ul style="list-style-type: none"> <li>• Grundlagen der Programmiersprache Python (Syntax, Operatoren, Kontrollstrukturen, Datenstrukturen, Ausnahmebehandlung)</li> <li>• Jupyter Notebook als Programmierumgebung</li> <li>• Techniken in Python zur Datenbehandlung (load, transform, reshape, merge, slice usw.) anhand aktueller Bibliotheken (z.B. NumPy, Pandas)</li> <li>• Arbeit mit einer Breite an Datentypen (numerische Daten, Zeitstempel, Bilder, Texte usw.), -formaten (CSV, JSON, XML usw.) sowie -sätzen</li> <li>• Grundlegende Verwendung von Datenvisualisierung anhand aktueller Bibliotheken (z.B. Matplotlib, plotly, seaborn)</li> <li>• Optional: Weitere vertiefende Themen (z.B. Einblick in Data Cleaning, Ausblick auf maschinelles Lernen mit scikit-learn, Ausblick auf Deep Learning mit TensorFlow,</li> <li>• Datenvisualisierung mit Web-App-orientierten Werkzeugen wie Streamlit)</li> </ul> <p>In den Übungen werden die Themen in Einzelarbeit am Rechner vertieft.</p>
<b>Literatur</b>	<p>Paul Barry: <i>Python von Kopf bis Fuß</i>. O'Reilly 2017.</p> <p>David Beazley, Brian K. Jones: <i>Python Cookbook</i>. O'Reilly 2013.</p> <p>Ethan Williams: <i>Python for Data Science: The Ultimate Beginners' Guide to Learning Python Data Science Step by Step</i>. Vom Autor ohne Verlag veröffentlicht 2019.</p> <p>Albert Sweigart: <i>Automate the Boring Stuff with Python, 2nd Edition: Practical Programming for Total Beginners</i>. No Scratch Press 2019.</p> <p>Joel Grus: <i>Data Science from Scratch: First Principles with Python</i>. O'Reilly 2019.</p> <p>Jake VanderPlas: <i>Python Data Science Handbook: Essential Tools for working with Data</i>. O'Reilly 2016.</p>

	Wes McKinney: <i>Python for Data Analysis: Data Wrangling with Pandas, Numpy, and Jupyter</i> . O'Reilly 2022. Weitere aktuelle Literatur wird in der Vorlesung angegeben.
Weitere Hinweise	Dieses Modul wird auf Deutsch angeboten.
Raumbedarf	