

Technology Arts Sciences TH Köln

TH KÖLN

INSTITUT FÜR INFORMATIK

Modulbeschreibungen Bachelor

22. Februar 2017



Inhaltsverzeichnis

1 Algorithmen und Programmierung 1	1
2 Algorithmen und Programmierung 2	3
3 Audiovisuelles Medienprojekt	5
4 Bachelorarbeit	9
5 Bachelor Kolloquium	11
6 Betriebssysteme und verteilte Systeme	13
7 BWL I - Grundlagen	16
8 Datenbanken 1	18
9 Einführung in Betriebssysteme und Rechnerarchitektur	21
10 Einführung in die Medieninformatik	23
11 Entwicklungsprojekt	25
12 Grundlagen des Web	27
13 Kommunikationstechnik und Netze	30
14 Mathematik 1	32
15 Mathematik 2	34
16 Mensch-Computer Interaktion	36
17 Informatik, Recht und Gesellschaft	40
18 Paradigmen der Programmierung	44
19 Praxisprojekt	47



20 Praxisprojektseminar	49
21 Projektmanagement	51
22 Empirische Forschungsmethoden in der Informatik	54
23 Projekt Social Computing	57
24 Gamification	59
25 Soziotechnische Systeme	62
26 Screendesign	65
27 Softwaretechnik	68
28 Theoretische Informatik 1	71
29 Theoretische Informatik 2	75
30 Visual Computing	79
31 Web Development	91
32 Social Computing	96
33 Wahlpflichtmodul	99



Kapitel 1

Algorithmen und Programmierung 1

Angestrebte Lernergebnisse:

Die Studierenden sollen

- die Prinzipien der Objektorientierung verstanden haben und einfache Softwareeinheiten unter Beachtung dieser Prinzipien erstellen können
- die wichtigsten Eigenschaften der Programmiersprachen C und Java sowie der Java-Entwicklungsumgebung kennen
- einfache Algorithmen beurteilen, vorhandene Bibliothekslösungen angemessen einsetzen und einfache Algorithmen selbstständig entwickeln können.

Inhalt:

- objektorientierte u. prozedurale Programmierung in C und Java
- Algorithmen mit elementaren Datentypen
- Rekursion in C

Literatur:

- Tutorium unter <http://www.hb02.de/ap1>



Kapitel 2

Algorithmen und Programmierung 2

Angestrebte Lernergebnisse:

Die Studierende sollen Objektorientierung, die Prinzipien der Algorithmenentwicklung und grundlegende Algorithmen verstehen und die Grundstrukturen der Java-Bibliothek anwenden können.

Inhalt:

- Basisalgorithmen: Suchen u. Sortieren
- Datenstrukturen
- Dictionaries
- Methodik des objektorientierten Programmierens

Literatur:

- Vorlesungsunterlagen: Foliensammlung, ausformuliertes Skript, Beispiellösungen, Verweise auf Onlinedokumente
- J. Bishop

Java Lernen

- R. Sedgewick



Algorithmen in Java

- J. Barnes, M. Kölling,

Java Lernen mit BlueJ



Kapitel 3

Audiovisuelles Medienprojekt

Lehrform/SWS:

4 SWS: Vorlesung 2 SWS; Projekt 2 SWS

Arbeitsaufwand:

Gesamtaufwand 150 h, davon

- 36 h Vorlesung
- 36 h Projektarbeit
- 78 h Selbststudium

Angestrebte Lernergebnisse:

Die praktische Umsetzung des Vorlesungsstoffes, die Kommunikation und Zusammenarbeit im Team über Themenbereiche dieses Faches und die Präsentation von eigenen Projekten und Untersuchungsergebnissen sind die Lernziele des Moduls „Audiovisuelles Medienprojekt“. Neben dieser formulierten Fachkompetenz, Methodenkompetenz und Kommunikationskompetenz stehen gerade die sogenannten Softskills Teamfähigkeit und Kommunikationsfähigkeit im Focus der Ausbildung in diesem Modul.

Um mit der sich hieraus entwickelnden erhöhten Komplexität der Fragestellungen kompetent umgehen zu können, wird in diesem Modul inhaltlich auf die technischen Grundlagen der au-



audiovisuellen Produktion besonders eingegangen. Die thematische Gewichtung der Inhalte ist im Hinblick auf die Vermittlung der Befähigung ausgerichtet, selbstständig in der audiovisuellen Produktion auftretende Problemstellungen lösen zu können und die verwendeten technischen Werkzeuge, wie Videokamera, Tonaufnahmegeräte und Schnittsysteme technisch richtig und gestalterisch aussagekräftig einzusetzen. Über diese Methodenkompetenz hinaus wird in Filmanalysen und einer kritischen Betrachtung medialer Ereignisse die Fähigkeit zur Reflexion vermittelt. Dabei spielt die Begründung der Auswahl bestimmter Medientechnologien und deren Einsatz in der Medienproduktion eine wichtige Rolle.

Die Studierenden kennen die grundlegenden Erzählformen audiovisueller Medien und haben folgende Fertigkeiten: Sie können eigene audiovisuelle Erzählformen auf der Basis klassischer Erzählmuster entwickeln und sind befähigt zur Analyse, Diskussion und zur kritischen Betrachtung audiovisueller Medieninhalte.

Pragmatisches Ziel ist es, in den unterschiedlichsten Berufsfeldern digitaler audiovisueller Medien die Entwicklung und den Einsatz audiovisuellen Content zu beraten, zu planen, durchzuführen oder zu verantworten.

Inhalt:

Im Mittelpunkt dieses Moduls steht die digitale audiovisuelle Medienproduktion.

Die Projektarbeit gliedert sich dabei in die selbstständige Entwicklung, Ausarbeitung und Präsentation eines Filmthemas, in die praktische Umsetzung in einem Filmprojekt und in die Nachbearbeitung und Montage in einer dramaturgischen Erzählform.

Begleitend zu der Produktion werden folgende fachspezifischen Inhalte thematisiert und in Übungsaufgaben vertieft:

- Video- und Audioaufnahmetechnik
- Filmsprache
- Lichtsetzung
- Tonaufnahme
- Dokumentarfilm und Interview
- Dramaturgie
- Schnitt und Montage



Studien-/Prüfungsleistungen:

Projektarbeit und schriftliche Ausarbeitung

Medienformen:

- Beamer-gestützte Vorlesungen (Folien in elektronischer Form);
- Beispiele aus verschiedenen Medien in elektronischer Form
- Projektarbeit in Teams, um die erlernten Methoden und Techniken einzuüben und zu vertiefen;

Literatur:

- James Monaco, Film verstehen

Rowolth Taschenbuch Verlag Hamburg, 1980,

ISBN 3-499-162717

- Syd Field, Drehbuchschreiben für Film und Fernsehen,

München 2003, ISBN 354836473X

- Steven D. Katz, Die Richtige Einstellung

Zweitausendeins, Frankfurt a.M.1998,ISBN 3-86150-229-1

- David Lewis Yewdall, Practical Art of Motion Picture Sound

Focal Press, USA 2003, ISBN 0-240-80525-9

- Hans Kornacher & Manfred Stross,

Dokumentarisches Videofilmen

Augustus Verlag, Augsburg, 1992, ISBN 3-8043-5474-2

- Hans Beller Hg., Handbuch der Filmmontage,



München: TR-Verlagsunion, 1993, ISBN 3-8058-2357-6

- Karel Reisz, Gavin Millar,

Geschichte und Technik der Filmmontage,

München: Film- und Landpresse, 1988, ISBN 3-88690-071-1

- Chris Vogler,

Die Reise des Drehbuchschreibens, Verlag Zweitausendeins



Kapitel 4

Bachelorarbeit

Lehrform/SWS:

Angeleitetes, eigenverantwortliches Arbeiten

Arbeitsaufwand:

360 Stunden

Angestrebte Lernergebnisse:

Die Bachelorarbeit soll zeigen, dass der Prüfling befähigt ist, innerhalb einer vorgegebenen Frist eine praxisorientierte Aufgabe aus seinem Fachgebiet sowohl in ihren fachlichen Einzelheiten als auch in den fachübergreifenden Zusammenhängen nach wissenschaftlichen, fachpraktischen und gestalterischen Methoden selbständig zu bearbeiten. Die Bachelorarbeit ist in der Regel eine eigenständige Untersuchung mit einer Aufgabenstellung aus der Medieninformatik und einer ausführlichen Beschreibung und Erläuterung ihrer Lösung. In fachlich geeigneten Fällen kann sie auch eine schriftliche Hausarbeit mit fachliterarischem Inhalt sein.

Inhalt:

Selbstständiges wissenschaftliches, fachpraktisches und gestalterisches Bearbeiten einer Aufgabenstellung.



Kapitel 5

Bachelor Kolloquium

Lehrform/SWS:

Angeleitetes, eigenverantwortliches Arbeiten

Arbeitsaufwand:

90 Stunden

Angestrebte Lernergebnisse:

Das Kolloquium dient der Feststellung, ob der Prüfling befähigt ist, die Ergebnisse der Bachelorarbeit, ihre fachlichen Grundlagen, ihre fachübergreifenden Zusammenhänge und ihre außerfachlichen Bezüge mündlich darzustellen und selbständig zu begründen und ihre Bedeutung für die Praxis einzuschätzen. Dabei soll auch die Bearbeitung des Themas der Bachelorarbeit mit dem Prüfling erörtert werden.

Inhalt:

Vortrag über das Thema der Bachelorarbeit, Fachdiskussion und mündliche Verteidigung der Arbeit im Kontext der Medieninformatik.



Studien-/Prüfungsleistungen:

mündliche Prüfung, Vortrag



Kapitel 6

Betriebssysteme und verteilte Systeme

Kurzbeschreibung

Systemprogrammierung am Beispiel von UNIX.

Lehrform/SWS:

4 SWS: Vorlesung 2 SWS; Praktikum 2 SWS

Arbeitsaufwand:

- 36 h Vorlesung
- 36 h Praktikum
- 78 h Selbststudium

Angestrebte Lernergebnisse:

Die Studierenden sollen:

- die Prinzipien und Mechanismen von Betriebssystemen und verteilten Systemen am Beispiel von UNIX verstanden haben,



-
- in der Lage sein, selbstständig Systemprogramme zu schreiben und Betriebssystemstrukturen zu bewerten und
 - die Mechanismen zur Implementierung verteilter Anwendungen anwenden können.

Inhalt:

Systemprogrammierung am Beispiel von UNIX:

- Shell-Programmierung
- Prozess-Modelle
- Prozess-Erzeugung und Synchronisation
- UNIX-Prozesse und elementare Synchronisation
- Pipes
- Shared Memory
- Synchronisationsprimitive für den wechselseitigen Ausschluss
- Semaphoren
- Nachrichtenwarteschlangen
- Dateisysteme
- TCP/IP
- Sockets
- Remote Procedure Call
- Strategien zum Scheduling und zur Speicherverwaltung
- Klassische Synchronisationsprobleme

Studien-/Prüfungsleistungen:

Klausur sowie erfolgreiche Teilnahme am Praktikum als Prüfungsvorleistung.



Medienformen:

Foliensammlung, ausformuliertes Skript, Beispiellösungen

Literatur:

- Tanenbaum, A. S.: „Moderne Betriebssysteme“
- Brown, C.: „Programmieren verteilter UNIX-Anwendungen“
- Kernighan, B. W., Pike, R.: „Der UNIX-Werkzeugkasten“
- Ehses, E., Köhler, L., Stenzel, H., Victor, F. „Betriebssysteme: Ein Lehrbuch mit Übungen zur Systemprogrammierung in UNIX/Linux“



Kapitel 7

BWL I - Grundlagen

Lehrform/SWS:

4 SWS: Vorlesung 2 SWS; Übung 2 SWS

Arbeitsaufwand:

Gesamtaufwand 120 h, davon

- 30 h Vorlesung
- 30 h Übung
- 60 h Selbstlernphase

Angestrebte Lernergebnisse:

Die Studierenden verstehen die wichtigsten Entscheidungsbereiche wirtschaftlichen Handelns und können diese anwenden. Sie können grundlegenden Entscheidungen im Rahmen einer Unternehmensgründung, die Aufgaben der Unternehmensführung wie die Konzeption einer tragfähigen Strategie, und die Aufgaben der Teilbereiche Produktion, Absatz und Marketing sowie Investition und Finanzierung beschreiben und beurteilen. Investitionsentscheidungen können die Studierenden informationsgestützt treffen indem Sie die Kalkulationsverfahren der Investitionsrechnung anwenden und auswerten. Die Veranstaltung bereitet die Studierenden für weitere BWL-Veranstaltungen Ihres Studiums, sowie darauf, in ihrem Berufsleben wirtschaftliche Konzepte im Unternehmenskontext anzuwenden, vor.



Inhalt:

- Grundlagen
- Unternehmensführung 1: Ziele, Planung und Entscheidung
- Investition und Finanzierung
- Unternehmensführung 2: Ausführung und Kontrolle
- Konstitutive Entscheidungen
- Produktion
- Absatz und Marketing

Studien-/Prüfungsleistungen:

schriftliche Klausur

Literatur:

- Wöhe (2016): Einführung in die Allgemeine Betriebswirtschaftslehre, 26. Aufl.



Kapitel 8

Datenbanken 1

Angestrebte Lernergebnisse:

Die Studierenden sollen

- über ein einheitliches konsistentes Begriffsgebäude bezüglich der Datenbankthematik verfügen,
- die theoretischen Grundlagen von Datenbanksystemen am Beispiel relationaler und objektrelationaler Datenbanksysteme verstanden haben, insbesondere die relationale Algebra, die Normalisierung sowie funktionale Abhängigkeiten
- in der Lage sein, diese Erkenntnisse im Rahmen der Modellierung und Implementierung von Datenbankschemata praktisch anzuwenden,
- komplexere Datenbankabfragen, Datendefinitionen und Datenänderungen über SQL programmieren können
- ein SQL-Statement tunen können
- mit dem Transaktionsbegriff, der Mehrbenutzersynchronisation und Verfahren zur Fehlererholung sowie zur Sicherung der Datenintegrität vertraut sein

Inhalt:

- Grundbegriffe und Architektur von Datenbanken
- Ein Vorgehensmodell zur Erstellung eines Datenbanksystems



-
- Grundlagen des relationalen Modells
 - Relationale Algebra
 - Anfrageoptimierung
 - Funktionale Abhängigkeiten
 - Datenintegrität
 - Normalisierung
 - Datenmodellierung (Entity Relationship Modell) und Implementierung am Beispiel eines relationalen Datenbanksystems
 - Datenbanksprache SQL: DDL, DML, DAL, Integritätsbedingungen und Constraints unter dem jeweils aktuellen SQL-Standard, zur Zeit SQL2013
 - Transaktionskonzepte, Mehrbenutzersynchronisation, Fehlererholung und Datensicherheit

Studien-/Prüfungsleistungen:

Klausur, deren Voraussetzung das bestandene Praktikum ist, und semesterbegleitende Multiple-Choice-Tests mit Punkten für die Klausur

Medienformen:

- Folien gestützter Vortrag - aber nur sehr selten
- I.d.R. erarbeiten der Theorie anhand von überschaubaren Problemstellungen und deren in der Veranstaltung entwickelten Lösungen - hauptsächliches Vorgehen
- Fragen der Studierenden beantworten - sehr erwünscht!
- Ilias zur Bereitstellung aller Informationen (Aktuelles, Links, Folien, Praktikums-/Übungsaufgaben, wie auch Lösungen)
- edb, die DB-eLearning-Plattform der TH Köln
- DB-Wiki, das Online Lexikon für Datenbank-Themen



Literatur:

- Elmasri, R.; Navathe, S. B.: Grundlagen von Datenbanksystemen. Pearson-Studium. 2009
- Faeskorn-Woyke, H.; Bertelsmeier, B.; Riemer, P.; Bauer, E.: Datenbanksysteme - Theorie und Praxis mit SQL2003, Oracle und MySQL. Pearson-Studium. 2. Aufl. 2011
- Kemper, A.; Eickler, A.: Datenbanksysteme – Eine Einführung. Oldenbourg-Verlag, 2015
- Saake, G., Sattler, K.-U.; Heuer, A.: Datenbanken - Konzepte und Sprachen. Mitp/bhv, 2013
- Vossen, G.: Datenmodelle, Datenbanksprachen, Datenbankmanagementsysteme. Oldenbourg-Verlag, 2008



Kapitel 9

Einführung in Betriebssysteme und Rechnerarchitektur

Angestrebte Lernergebnisse:

Die Studierenden sollen die Basiskonzepte und Grundlagen der Betriebssysteme und der Rechnerarchitektur kennen und verstehen, sowie ein einheitliches konsistentes Begriffsgebäude zu, teilweise aus der persönlichen Praxis bekannten, Sachverhalten der IT aufbauen

Inhalt:

- Betriebssysteme aus Nutzersicht: Dateisysteme, Parallele Prozesse, Sicherheit in Betriebssystemen
- bei Rechnerkomponenten: grundlegende Architekturen, Darstellung von Daten, interne Bussysteme, Prozessoren, Festplatten, Peripherieschnittstellen, Parallelrechner

Literatur:

- Vorlesungsunterlagen: kommentierte Foliensammlung
- Tanenbaum: „Rechnerarchitektur“
- Tanenbaum: „Modern Operating Systems“



Kapitel 10

Einführung in die Medieninformatik

Lehrform/SWS:

Vorlesung mit eingebetteten Übungselementen

Arbeitsaufwand:

150 Stunden

Angestrebte Lernergebnisse:

Die Studierenden können die inhaltlichen Ausrichtungen und die Zielsetzungen der Lehr- und Anwendungsdisziplin Medieninformatik benennen und gegenüber verwandten oder ähnlichen Disziplinen abgrenzen.

Die Studierenden kennen Grundkonzepte der Informatik (z.B. Anforderungen) sowie audiovisueller und interaktiver Medientechnologien, kennen architekturelle Alternativen interaktiver Systeme und kennen Gestaltungsdimensionen für deren Informations- und Kommunikationsinhalte. Die Studierenden sind sensibilisiert für Modellierungs- und Entwicklungsaufgaben von medienbasierten Software-Systemen zur Unterstützung menschlichen Handelns in betrieblichen und privaten Kontexten.

Sie kennen grundlegende Konzepte, Prozesse/Verfahren und Modelle der Medieninformatik und haben erste Projekterfahrungen gesammelt. Sie können Systemkonzeptionen, zugehörige Modellierungen, Abwägungen und Artefakte für ein Fachpublikum angemessen dokumentieren und mittels verschiedener medialer Formen kommunizieren.



Inhalt:

Workshops zu grundlegenden projektrelevanten Themenfeldern (wie: Datenmodellierung, Pseudo-Code, Kommunikation in verteilten medialen Systemen, Visual Thinking, Storytelling, Anforderungen) und deren Anwendung, illustriert anhand von Fallstudien.

Teambasiertes Projekt, welches ausgehend von Kontextszenarien eine (oder mehrere) Problemstellung(en) umreist, zu dem Lösungen konzipiert und prototypisch umgesetzt, dokumentiert und einem Fachpublikum präsentiert werden müssen.

Studien-/Prüfungsleistungen:

mündliche Prüfung, Projektpräsentation, schriftliche Ausarbeitung

Medienformen:

- Beamer-gestützte Vorlesungen (Folien in elektronischer Form)
- Vorträge
- verschiedene Präsentationsmaterialien (Whiteboard, Poster, etc.)
- Einsatz von Bild- und Videobearbeitungssoftware
- Umgang mit Kameras im Projektteil

Literatur:

- Michael Herczeg: Einführung in die Medieninformatik, Oldenbourg Verlag, 2006, ISBN: 3-486-581-031
- Chris Rupp et al: Requirements-Engineering und -Management: Aus der Praxis von klassisch bis agil, Carl Hanser Verlag; 6-te Auflage, 2014, ISBN-10: 3446438939



Kapitel 11

Entwicklungsprojekt

Lehrform/SWS:

Projekt

Arbeitsaufwand:

300 h

Angestrebte Lernergebnisse:

Die Studierenden sollen vertiefende Kenntnisse in die Methoden und Techniken aus zwei ausgewählten Modulen aus den ersten vier Fachsemestern des Studiums erlangen und diese in der Konzeption und prototypischen Realisierung eines interaktiven Systems oder Mediums anwenden. Dadurch sollen sie eigene Erfahrungen in der Projektabwicklung mit Medieninformatik-spezifischen Fragestellungen und in der Teamarbeit sammeln und eine reflektierend-kritische Haltung zu methodischen Ansätzen und Entwicklungsmodellen entwickeln. Ziel ist es eine, mit eigenen praktischen Erfahrungen fundierte Methodenkompetenz zu erlangen.

Die Studierenden sollen darüberhinaus lernen, die Vorgehensweise und die Ergebnisse ihres Projektes in einem kritischen Diskurs vor einem Fachpublikum zu vertreten, um in der Berufspraxis ihre Herangehensweise und Projektergebnisse vertreten zu können.



Inhalt:

Die Projekte werden in Teams durchgeführt. Zunächst werden von den Teams zwei Module aus den ersten vier Fachsemestern gewählt, welche die fachlichen Perspektiven für die Vertiefung bestimmen. In Absprache mit den Lehrenden werden dann Projektziele festgelegt.

Aus dem Methoden- und Technikkatalog (siehe Vorlesungen zu den Lehrbereichen) wird in Absprache mit den Lehrenden eine Auswahl der einzusetzenden Entwicklungstechniken und -methoden sowie der einzuhaltenden Entwicklungsmodelle getroffen und Qualitätssicherungsmaßnahmen und das Prozessmanagement definiert.

Die Lehrenden bieten dann während der Projektdurchführung Beratung bzgl. des adäquaten Einsatzes der gewählten Methoden und Techniken. Zwischenstände des Projektes werden zu definierten Meilensteinen von den Projektteams präsentiert. Die Präsentation der Projektergebnisse findet in einem Plenum mit kritischer Diskussion der Methoden und Ergebnisse statt.

Studien-/Prüfungsleistungen:

Die Projektergebnis, bestehend aus Prototyp und Dokumentation, geht mit 80% in die Projektnote ein, die Kommunikation der Zwischenergebnisse und des Endergebnisses mit 20%.



Kapitel 12

Grundlagen des Web

Kurzbeschreibung

In der Veranstaltung werden wesentliche Grundideen, Interaktionsprinzipien, Contentarchitekturen und Sicherheitsmechanismen eingeführt, die das Web als Medium konstituieren.

Lehrform/SWS:

4 SWS: Vorlesung 2 SWS; Seminar 2 SWS

Arbeitsaufwand:

150 Stunden

Angestrebte Lernergebnisse:

Die Studierenden

- kennen wesentliche Grundideen, Interaktionsprinzipien, Contentarchitekturen und Sicherheitsmechanismen, die das Web als Medium konstituieren und
- können moderne Webanwendungen auf der Basis von Fachbegriffen analysieren und einordnen



-
- um kompetent am fachlichen Diskurs über Eigenschaften, Auswirkungen und Gestaltungsalternativen von Web Anwendungen teilnehmen zu können.

Inhalt:

- Web Architektur des W3C
- Offenheit und Verwendung von Standards als Prinzip
- Interaktionsformen: Synchrone Interaktion auf der Basis von REST, asynchrone Interaktion mit Publish/Subscribe
- Fallstudien: Cloudservices für verteilte Anwendungen - z.B. Amazon Web Services, Google Firebase
- Ausgewählte Sicherheitsmechanismen im Web
- Inhaltsarchitekturen: XML, JSON, Microformate, RDFa

Die Inhalte werden als Vorlesung vermittelt. In dem begleitenden Seminar werden die Konzepte mittels Fallstudien anwendungsbezogen analysiert und diskutiert.

Studien-/Prüfungsleistungen:

Mündliche Prüfung

Medienformen:

- Folienpräsentation
- Auschnitte aus der Literatur als Leseaufgaben und Fallstudien

Literatur:

- Randy Conolly, Richard Hoar: Fundamentals of Web Development, Pearson Publishing 2015



-
- Hugh Taylor et al.: Event-Driven Architecture - How SOA Enables the Real-Time Enterprise, Addison-Wesley 2009
 - Webber: REST in Practice, OReilly 2011
 - Sam Newman: Building Micro Services, OReilly 2015
 - James Governor et al.: Web 2.0 Architectures, OReilly 2009
 - Rajkumar Buyya (ed.): Internet of Things: Principles and Paradigms, Morgan Kaufmann 2016



Kapitel 13

Kommunikationstechnik und Netze

Lehrform/SWS:

Vorlesung, Praktikum

Angestrebte Lernergebnisse:

Die Studierenden sollen

- Prinzipien und Grundlagen von technischen Kommunikationsvorgängen kennen lernen,
- Protokolle als wesentliche Grundlage der Kommunikationstechnik im Detail verstehen (Internet-Protokolle, Multimedia-Protokolle, TK-Protokolle, Dienste)
- Einsatz und Nutzung von Kommunikationstechnik praxistypisch kennen lernen,
- in der Lage sein, selbstständig Netzstrukturen zu bewerten, Netze zu analysieren und zu konzipieren (unter Anwendung von Netzanalysewerkzeugen und -methoden).

Inhalt:

Grundbegriffe und Grundlagen, Kommunikationssysteme (Modelle, Grundbegriffe), Protokolle, Schnittstellen, Dienste, Architekturmodelle (OSI-Referenzmodell, TCP/IP-Protokollfamilie), Standardisierung, TCP/IP-Protokollfamilie als Grundlage des Internet, Schichtenmodell und Protokolle im Detail, Adressierung, ausgewählte Anwendungen, Klassifizierung von Netzen / Topologien / Technologien, Wegewahl / Vermittlung / Routing, Vermittlungsprinzipien, Routing-Verfahren



und Protokolle, Internet-spezifische Verfahren, Multimedia-Netze, Dienstgüte, Internet-Telefonie, Realisierung von Multimedia-Netzen, Netzsicherheit, grundlegende Begriffe der „IT-Sicherheit“, typische Bedrohungen in Netzen, Beispielszenarien

Medienformen:

- Vorlesung im Hörsaal (PowerPoint und Beamer)
- Praktikum an Rechnern des KTDS-Labors; Ressourcen: Netzanalysesoftware, div. Netzüberwachungssoftware, E-Mail-Server und -Clients, DNS-Server, ggf. weitere Server-Implementierungen

Literatur:

- Wird in der Veranstaltung bekannt gegeben



Kapitel 14

Mathematik 1

Lehrform/SWS:

6 SWS: Vorlesung 3 SWS; Praktikum 1 SWS; Übung 2 SWS

Angestrebte Lernergebnisse:

Die Studierenden sollen die Fähigkeiten zur Analyse realer oder geplanter Systeme entwickeln, indem sie praktische Aufgabenstellungen aus dem Informatik-Umfeld in mathematische Strukturen abstrahieren und lernen, selbstständig die Modellfindung und die Ergebnisbeurteilung vorzunehmen. Dabei sollen die Anwendungsbezüge der Mathematik deutlich werden, z.B. die Bedeutung funktionaler Beziehungen für kontinuierliche Zusammenhänge, die lineare Algebra z.B. als Grundlage der grafischen Datenverarbeitung und die Analysis zur Verarbeitung von Signalen und zur Lösung von mathematischen Modellen.

Inhalt:

- Grundlagen
- Folgen
- Funktionen
- Differenzialrechnung (1 Veränderliche)
- Integralrechnung



-
- Lineare Algebra

Studien-/Prüfungsleistungen:

Klausur (60 min) sowie erfolgreiche Teilnahme am Praktikum als Zulassungsvoraussetzung

Literatur:

- Skript unter www.gm.fh-koeln.de/~kone
- Teschl, Gerald und Teschl, Susanne: "Mathematik für Informatiker", Springer Verlag, 4. Auflage, 2013
- Hartmann, Peter: "Mathematik für Informatiker-Ein praxisbezogenes Lehrbuch" Vieweg Verlag, 475 Seiten, 3. Auflage 2006
- Papula, Lothar: "Mathematik für Ingenieure und Naturwissenschaftler" Vieweg Verlag, 14. Auflage, 2014
- Stingl, Mathematik für Fachhochschulen, Hanser 2003



Kapitel 15

Mathematik 2

Lehrform/SWS:

6 SWS: Vorlesung 3 SWS; Praktikum 1 SWS; Übung 2 SWS

Angestrebte Lernergebnisse:

Die Studierenden sollen die Fähigkeiten zur Analyse realer oder geplanter Systeme entwickeln, indem sie praktische Aufgabenstellungen aus dem Informatik-Umfeld in mathematische Strukturen abstrahieren und lernen, selbstständig die Modellfindung und die Ergebnisbeurteilung vorzunehmen. Dabei sollen die Anwendungsbezüge der Mathematik deutlich werden, z.B. die Beziehungen diskreter Strukturen wie der Graphen zu vielfältigen grundlegenden Datenstrukturen, die Statistik zur Deskription und Beurteilung von Beobachtungen und die Analysis zur Verarbeitung von Signalen und zur Lösung von mathematischen Modellen.

Inhalt:

- Mehrdimensionale Differenzialrechnung,
- Graphentheorie,
- Kombinatorik, Wahrscheinlichkeitsrechnung und Statistik,
- Komplexe Zahlen,
- Differentialgleichungen.



Studien-/Prüfungsleistungen:

Klausur (60 min) sowie erfolgreiche Teilnahme am Praktikum als Zulassungsvoraussetzung

Literatur:

- s. Literaturliste auf der Homepage www.gm.fh-koeln.de/~konen
- Skript unter www.gm.fh-koeln.de/~konen/Mathe2-SS



Kapitel 16

Mensch-Computer Interaktion

Lehrform/SWS:

Vorlesung und Übung

Arbeitsaufwand:

Gesamtaufwand 300 h, davon

- 65h Vorlesung
- 65h Übung
- 170 h Selbststudium

Angestrebte Lernergebnisse:

- Die Studierenden erwerben Grundkenntnisse in kognitions-, arbeits- und organisationspsychologischen Grundkonzepten und können diese auf Problemstellungen, im Kontext der Mensch-Computer Interaktion, anwenden.
- Die Studierenden kennen Modelle, Methoden, Arbeits- und Dokumentationstechniken der Mensch-Computer Interaktion, können sie anwenden, kritisch diskutieren und für konkrete Aktivitäten in Entwicklungsprojekten unter Abwägung der Alternativen auswählen.



-
- Sie kennen relevante internationale Normen und Standards, können sie anwenden und kritisch diskutieren .
 - Sie kennen methodische Ansätze benutzer- oder benutzungsorientierter Entwicklungsprozesse und können diese systematisch und iterativ auf die Konzeption, Realisation, Evaluation und das Redesign von interaktiven Systemen anwenden.
 - Zudem kennen sie Konzepte und Vorgehensmodelle für die Integration von Software- und Usability Engineering in einem Gesamtprozess und können diese in Entwicklungsprojekten anwenden.
 - Die Studierenden erlangen die Fähigkeit zum fachlichen Diskurs.

Inhalt:

- kognitionspsychologische Grundlagen
- Benutzermodellierung
- Tätigkeitsmodellierung
- Spezifikationsformen für Nutzungskontexte
- Spezifikation von Nutzungsanforderungen
- Interaktionsmodelle
- Interaktionsmodalitäten und –kodalitäten
- Vorgehensmodelle (human-centered, usability-engineering, usage-centered design)
- Design-Prinzipien, -Pattern, -Guidelines, -Styleguides
- Prototyping und Sketching
- Evaluation

Studien-/Prüfungsleistungen:

schriftliche Modulprüfung



Medienformen:

- Beamergestützte Vorlesung
- Case Studies
- Lehrfilme

Literatur:

- Dix, A.; Finlay, J.; Abowd, G. & Beale, R.: Human-Computer Interaction. Harlow, Pearson, 2004 (3rd ed.),
- Benyon, D., Turner, S. Turner, P. Designing Interactive Systems: People, Activities, Contexts, Technologies, Addison Wesley, 2005,
- Anderson, J.R.: Kognitive Psychologie. Heidelberg, Springer, 2001 (3. Auflage).
- Beyer H. & Holtzblatt K.: Contextual Design: Defining Customer-Centered Systems. San Francisco Morgan Kaufmann, 1997.
- Cockburn, A.: Writing Effective Use Cases. Boston, Addison-Wesley, 2000.
- Constantine, L.; Lockwood, L.: Software for Use, ACM Press, 1999.
- Dumas, J.S. & Redish, J.C.: A Practical Guide to Usability Testing. Exter, Intellect Books, 1999 (rev. edition).
- Hacker, W.: Allgemeine Arbeitspsychologie. Bern, Huber, 1998.
- Hackos, J. & Redish, J.: User and Task Analysis for Interface Design. New York, Wiley, 1998.
- Holtzblatt K.; Wendell, J.B. & Wood, S.: Rapid Contextual Design. A How-to Guide to Key Techniques for User-Centered Design. San Francisco, Morgan Kaufmann, 2005.
- Johnson, J.: GUI Bloopers. San Francisco, Morgan Kaufmann, 2000.
- Kulak, D. & Guiney, E.: Use Cases. Requirements in Context. Boston, Addison-Wesley, 2000.
- Mayhew, D.: The Usability Engineering Lifecycle. A Practitioner's Handbook for User Interface Design. San Francisco: Morgan Kaufmann, 1999.
- Nielsen, J. & Mack, R.L. (eds.): Usability Inspection Methods. New York, Wiley, 1994.



-
- Preece, J; Rogers, Y. & Sharp, H.: Interaction Design. Beyond Human-Computer Interaction. NewYork, Wiley, 2002.
 - Rosson, M.B. & Carroll, J.M.: Usability Engineering. Scenario-Based Development of Human-Computer Interaction. San Francisco, Morgan Kaufmann, 2002.
 - Snyder, C: Paper Prototyping. San Francisco, Morgan Kaufmann, 2003.
 - Ulich, E.: Arbeitspsychologie. Stuttgart, Schäffer-Poeschel, 2001 (5.Auflage).



Kapitel 17

Informatik, Recht und Gesellschaft

Lehrform/SWS:

4 SWS: Vorlesung 2 SWS; Übung 2 SWS

Arbeitsaufwand:

Gesamtaufwand: 150 h, davon

- 36 h Vorlesung
- 36 h Übung
- 78 h Selbststudium

Angestrebte Lernergebnisse:

Informatikerinnen und Informatiker analysieren und konstruieren sozio-technische Systeme und entwickeln dabei semiotische Artefakte wie z.B. Spezifikationen, Programme und Handbücher. Die entwickelten Systeme bilden einerseits soziale Wirklichkeit in vielfältiger Form ab und ändern andererseits diese Wirklichkeit durch ihren Einsatz.

Die Studierenden sollen befähigt werden

- die unterschiedlichen Wechselwirkungen zwischen Informatik-Systemen und ihrem Einsatzumfeld zu erkennen und zu bewerten,



- ethische und rechtliche Aspekte des Einsatzes von Informatik-Systemen zu charakterisieren
- ein kritisches Bewusstsein für die aktuellen Fragen des wechselseitigen Einflusses von Informatik und Gesellschaft zu entwickeln.
- die Grundbegriffe des deutschen Privatrechts zu verstehen
- sich im dazugehörigen Gesetzeswerk zu orientieren
- und insbesondere im Bereich des Vertragsrechts selbständige Lösungsvorschläge zu erarbeiten.

Inhalt:

Informatik und Gesellschaft

Die Wechselwirkungen zwischen den von Informatikern entwickelten Systemen und ihrem Einsatzumfeld werden in drei großen Themenblöcken behandelt:

- Informatik und soziale Kontexte
- Komplexität und Sicherheit in sozio-technischen Systemen
- Systemgestaltung und Verantwortung der Informatik.

Beispielhafte Inhalte:

- Geschichte der Informatik
- Bildung und Wissenschaft
- Wissenschaften und Gesellschaft
- Digitale Medien und Internet
- Datenschutz und Überwachungstechniken
- Informatik und Gestaltung
- partizipative Systemgestaltung
- Open Source
- Ethische Leitlinien für Informatiker



- Normen und Standards
- philosophische Aspekte der Informatik

Recht

- Einführung in das deutsche Privatrecht, insbesondere in das BGB.
- Schwerpunkt im Schuldrecht, hier insbesondere im Vertragsrecht.
- Besondere Aspekte des Verbraucherschutzes und der inhaltlichen Gestaltung von Verträgen.
- Im Allgemeinen Teil des BGB wird auf den Vertragsschluss, die Willenerklärung als rechtsgeschäftliches Gestaltungsmittel und die allgemeinen Anforderungen an die Vertragspartner eingegangen.

Studien-/Prüfungsleistungen:

Informatik und Gesellschaft

Präsentation im OpenSpace, Klausur (60 Min).

Recht

Klausur (60 Min.)

Medienformen:

Beamergestützte Vorträge



Literatur:

IUG

- Sara Baase: A Gift of Fire. Social, Legal, and Ethical Issues in Computing. Prentice Hall, Upper Saddle River, 1997
- A.F. Chalmers: Wege der Wissenschaft. 5. Aufl., Springer, Heidelberg, 2001
- D.M. Hester, P.J. Ford: Computers and Ethics in the Cyberage. Prentice Hall, Upper Saddle River, 2001
- P. Gola, C. Klug: Grundzüge des Datenschutzrechts. C.H. Beck, 2003
- M. Pierson, D. Seiler: Internet-Recht im Unternehmen. Beck-Rechtsberater im dtv, Deutscher Taschenbuch Verlag, München, 2002
- <http://www.gi-ev.de> Arbeitskreis Informatik und Verantwortung, Ethische Leitlinien der GI
- <http://www.bfd.bund.de> Der Bundesbeauftragte für den Datenschutz
- <http://www.aktiv.org/DVD> Deutsche Vereinigung für Datenschutz
- <http://www.big-brother-award.org> Überwachungsinformationen

Recht

- Bürgerliches Gesetzbuch in der aktuellen Taschenbuchausgabe des dtv

Fakultativ

- Eugen Klunziger, Einführung in das Bürgerliche Recht, Verlag Vahlen
- Norbert Ullrich, Wirtschaftsrecht für Betriebswirte, Verlag Neue Wirtschaftsbriefe



Kapitel 18

Paradigmen der Programmierung

Lehrform/SWS:

4 SWS: Vorlesung 2 SWS; Praktikum SWS; Übung 1SWS

Arbeitsaufwand:

Gesamtaufwand 150 h, davon

- 36 h Vorlesung
- 18 h Praktikum
- 18 h Übung
- 78 h Selbststudium

Angestrebte Lernergebnisse:

- Unterscheidung zwischen verschiedenen imperativen und deklarativen Programmierparadigmen kennen
- Einordnung der Anwendbarkeit unterschiedlicher Programmierkonzepte



Inhalt:

- Grundlagen von Programmiersprachen
- Vergleich imperativer und deklarativer Paradigmen
- prozedurale und objektorientierte Programmierung
- funktionale Programmierung
- Logikprogrammierung
- Nebenläufigkeit
- aspektorientierte Programmierung

Studien-/Prüfungsleistungen:

Klausur sowie erfolgreiche Teilnahme am Praktikum als Prüfungsvorleistung

Medienformen:

- Foliensammlung
- Skript
- Beispiellösungen

Literatur:

- Abelson, Sussman, Struktur und Interpretation von Computer Programmen, Springer-Verlag 2001
- W.F. Clocksin, C.S. Mellish, Programming in Prolog, Springer-Verlag 2003
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (2015). Design patterns: Entwurfsmuster als Elemente wiederverwendbarer objektorientierter Software. Frechen: Mitp.
- Odersky, Spoon, Venners, Programming in Scala, Artima Press 2011



-
- Goetz, Bloch, Bowbeer, Lea, Java-Concurrency in Practise, Addison Wesley 2006



Kapitel 19

Praxisprojekt

Lehrform/SWS:

Angeleitetes, eigenverantwortliches Arbeiten

Arbeitsaufwand:

300 h Projektarbeit

Angestrebte Lernergebnisse:

Die Studierenden

- können Methoden und Techniken, die sie im Studium erlernt haben, in realitätsnahen Projekten weitgehend selbstständig anwenden
- haben erste Erfahrungen mit der Selbststeuerung und proaktiven Kommunikation in einem Projekt mittlerer Größe und der Einordnung von Projektarbeit in betriebliche, gesellschaftliche und rechtliche Rahmenbedingungen gesammelt

Inhalt:

Modulinhalte des ersten bis fünften Semesters anhand von realen Anforderungen in einem praxisrelevanten Kontext anwenden und den Studierenden durch die Betreuung des Dozenten an



eine selbstständige Projektdurchführung und Kommunikation heranzuführen. Das Praxisprojekt kann entweder in einem Unternehmen oder in der Hochschule - dann eingebettet in Forschungsprojekte - erfolgen.

Studien-/Prüfungsleistungen:

Schriftliche Ausarbeitung, Projektdokumentation



Kapitel 20

Praxisprojektseminar

Lehrform/SWS:

Seminar

Arbeitsaufwand:

Gesamtaufwand 150 h, davon

- 32 h Seminar
- 118 h Selbststudium

Angestrebte Lernergebnisse:

Die Studierenden

- kennen Techniken wissenschaftlichen Arbeitens
- haben erste Erfahrungen mit aktiver Fachkommunikation gesammelt
- gewinnen einen ersten Überblick über das Spektrum von aktuellen Themen in der Medieninformatik
- können eigene Projektergebnisse vor einem Fachpublikum in Vortrag und Diskussion darstellen und verteidigen



Inhalt:

Das Praxisprojektseminar besteht aus

- Veranstaltungen in denen Techniken wissenschaftlichen Arbeitens vermittelt werden,
- Fachvorträgen von Studierenden über Ihre Projektergebnisse.

Studien-/Prüfungsleistungen:

Seminarvortrag zur Praxisprojektarbeit

Literatur:

- Rechenberg: Technisches Schreiben (nicht nur) für Informatiker, 2. Aufl, Hanser Verlag 2003
- M. Karmasin, R. Ribing: Die Gestaltung wissenschaftlicher Arbeiten, 2. Auflage WUV 2007



Kapitel 21

Projektmanagement

Kurzbeschreibung:

Managementaspekte der professionellen Entwicklung großer Softwaresysteme

Lehrform/SWS:

4 SWS: Vorlesung 2 SWS, Übung 1 SWS, Praktikum 1 SWS; max. 6 Studierende / Praktikumsteam;

Arbeitsaufwand:

Gesamtaufwand 150 h, davon

- 36 h Vorlesung
- 18 h Übung
- 18 h Praktikum
- 78 h Selbststudium

Angestrebte Lernergebnisse:

Die Studierenden sollen befähigt werden,



- die grundlegenden Aufgaben des Projektmanagements, insb. in IT-Projekten, zu charakterisieren und durchzuführen
- die Projektmanagement-Methoden, -Techniken und -Werkzeuge zielgerichtet einzusetzen
- die erforderlichen soziologischen und kommunikativen Aspekte zu berücksichtigen,

um, mit dem Ziel einer menschengerechten und soziologisch fundierten Menschenführung, eine wirkliche und optimale Produktivität bei komplexen Projekten erreichen zu können.

Inhalt:

Das Modul befasst sich mit den Managementaspekten der professionellen Entwicklung großer Softwaresysteme.

Der Vorlesungsteil des Moduls gliedert sich in folgende Kapitel:

- Überblick – Warum Projektmanagement?
- Teamarbeit und Menschenführung (Kommunikation und Führung)
- Kosten/Nutzen-Analysen und Entscheidungstechniken
- Projektorganisation und Projektplanung (Aufbauorganisation, Ablauforganisation, Prozessmodellierung, iterative und agile Vorgehensmodelle, Netzplantechnik)
- detaillierte Aufwandsschätzung und Projektcontrolling (Function Point Analysis, COCOMO, Risikomanagement, Projektpräsentationen)
- Inhalte von PM-BOK (Project Management - Body of Knowledge)
- Zusammenfassung und Prüfungsvorbereitung;

Damit die Studierenden die vorgestellten Methoden und Techniken zum Management von Softwareprojekten anwenden, sowie besser analysieren und bewerten können, werden in Projekt-Teams die in der Vorlesung vermittelten Inhalte anhand eines Fallbeispiels eingesetzt. Dazu bilden die Teilnehmenden Teams zu jeweils 6 Studierenden. Im Praktikum werden folgende Bereiche vertieft:

- Kosten- Nutzenrechnung, Entscheidungstechniken
- Aufbauorganisation
- Aufwandsschätzung (Function-Point-Analyse, COCOMO);
- Risikomanagement



- Ablauf- und Ressourcenplanung (Netzplantechnik, Einsatz von PM-Software wie z.B. MS-Project)

Studien-/Prüfungsleistungen:

- Praktikum-Ausarbeitung
- Vortrag
- Mündliche Prüfung.

Medienformen:

- Beamer-gestützte Vorlesungen (Folien in elektronischer Form im Netz);
- Vertiefende Unterlagen sowie aktuelle Artikel (in elektronischer Form im Netz);
- Praktika in Kleingruppen, um die erlernten Methoden und Techniken einzuüben und zu vertiefen (Seminarraum, Rechnerlabor);

Literatur:

- A. Buhl: Grundkurs Projektmanagement. Carl Hanser Verlag, München, 2004
- H.W. Wieczorrek, P. Mertens: Management von IT-Projekten Von der Planung zur Realisierung. 4. Aufl., Springer, Heidelberg, 2011
- C. Aichele, M. Schönberger: IT-Projektmanagement. Springer Vieweg, 2014
- A. Henrich: Management von Softwareprojekten. Oldenbourg Verlag, München, 2002
- H. Kerzner: Projektmanagement – Ein systemorientierter Ansatz. mitp-Verlag, Bonn, 2003
- T. DeMarco: Der Termin. Hanser, München, 1998



Kapitel 22

Empirische Forschungsmethoden in der Informatik

Lehrform/SWS:

4 SWS: Vorlesung 2 SWS; Seminar 2 SWS

Arbeitsaufwand:

Gesamtaufwand 150 h, davon

- 30 h Vorlesung
- 75 h Seminar
- 45 h Selbststudium

Angestrebte Lernergebnisse:

Die Studierenden sollen

- die unterschiedlichen Herangehensweisen quantitativer und qualitativer Forschungsmethoden verstehen. Darüber hinaus sollen
- ausgewählte quantitative und qualitative Methoden angewendet werden können.



- in der Lage sein, einfache Forschungsdesigns zu entwickeln und nach wissenschaftlichen Standards durchzuführen.

Inhalt:

Wissenschaftstheoretische Grundlagen

- Induktion, Deduktion, Abduktion
- Unterschied zwischen quantitativer und qualitativer Forschung
- Interviews gestalten, durchführen und auswerten
- Beobachtung
- Ethnographische Methoden
- Hypothesengewinnung und Theoriebildung
- Experimental-Designs
- Statistische Verfahren für quantitative Forschung
- Aussagekraft der Ergebnisse (statistische Signifikanz, interne und externe Validität)

Studien-/Prüfungsleistungen:

Abschlusspräsentation und schriftliche Ausarbeitung

Medienformen:

- Beamer-gestützte Vorlesungen (Folien in elektronischer Form)
- Screencasts und Handouts



Literatur:

- Döring, N. & Bortz, J. (2015). Forschungsmethoden und Evaluation: Für Human- und Sozialwissenschaftler. Berlin [u.a.]: Springer.
- Flick, U. (2011). Qualitative Sozialforschung: Eine Einführung. Reinbek bei Hamburg: Rowohlt-Taschenbuch-Verl.
- Popper, K. R. (1972). The logic of scientific discovery. London: Hutchinson.
- Schnädelbach, H. (2002). Erkenntnistheorie zur Einführung. Zur Einführung, 268. Hamburg: Junius.
- Westermann, R. (2000). Wissenschaftstheorie und Experimentalmethodik: Ein Lehrbuch zur psychologischen Methodenlehre. Göttingen [u.a.]: Hogrefe, Verl. für Psychologie.



Kapitel 23

Projekt Social Computing

Lehrform/SWS:

4 SWS Projekt

Arbeitsaufwand:

150h Projekt

Angestrebte Lernergebnisse:

TODO

Inhalt:

TODO

Studien-/Prüfungsleistungen:

TODO



Medienformen:

TODO

Literatur:

TODO



Kapitel 24

Gamification

Lehrform/SWS:

4 SWS: Vorlesung 1 SWS; Projekt 3 SWS

Arbeitsaufwand: 150 h, davon 30h Vorlesung, 90h
Projekt

Gesamtaufwand 150 h, davon

- 30 h Vorlesung
- 90 h Projekt

Angestrebte Lernergebnisse:

Die Studierenden sollen in der Lage sein, die Möglichkeiten und Grenzen des Gamification-Ansatzes, sowohl für die analoge als auch die digitale Welt, einordnen zu können. Die verschiedenen Stufen der Gamification sollen verstanden und die Maßnahmen in Gestaltungsprozessen eingesetzt werden. Die Studierenden sollen die psychologischen Grundlagen verstehen und die gesellschaftlichen Implikationen bewerten können. Die Analyse von Regeln und Prozessen und daraus abgeleitete Gamification-Maßnahmen sowie die Entwicklung von Serious Games sollen praktisch umgesetzt werden, z.B. für Lernanwendungen, Online-Communities oder soziale Dienste.



Inhalt:

- Grundelemente der Gamification
- Stufen der Gamification
- Ludifikation
- Historische Grundlagen
- Psychologische Grundlagen
- Gesellschaftliche Einordnung
- Einsatzgebiete
- Planung und Realisierung von Gamification
- Gestaltungsregeln
- Serious Games

Studien-/Prüfungsleistungen:

Projektausarbeitung

Medienformen:

- Folien
- Beispielmaterien
- Arbeit im Innovationsraum mit digitalen Whiteboards, Spiel-Arcade, Tablets und Gestaltungsmaterialien

Literatur:

- DeKoven, B., & MIT Press. (2013). The well-played game: A player's philosophy. Cambridge: The MIT Press.



-
- Fullerton, T., Swain, C., & Hoffman, S. (2008). Game design workshop: A playcentric approach to creating innovative games. Amsterdam: Elsevier Morgan Kaufmann.
 - Koster, R. (2013). Theory of Fun for Game Design. Sebastopol: O'Reilly.
 - Salen, K., & Zimmerman, E. (2007). Rules of play: Game design fundamentals. Cambridge, Mass. [u.a.: The MIT Press.



Kapitel 25

Soziotechnische Systeme

Lehrform/SWS:

4 SWS: Vorlesung 2 SWS; Seminar 2 SWS

Arbeitsaufwand:

Gesamtaufwand 150 h, davon

- 45 h Vorlesung
- 60 h Projekt
- 45h Selbststudium

Angestrebte Lernergebnisse:

Die Studierenden sollen das komplexe Wechselspiel zwischen informationstechnischen Systemen und gesellschaftlichen Normen und Prozessen verstehen, analysieren und einordnen können. Sie sollen in der Lage sein, Systeme nach ethischen, psychologischen und soziologischen Kriterien zu gestalten und die Auswirkungen einschätzen zu können. Die grundlegenden Modelle der verschiedenen Disziplinen sollen bekannt und verstanden werden.



Inhalt:

- Modelle der Sozioinformatik
- E-Learning
- Soziale und technische Systeme
- Gestaltungsprinzipien für soziotechnische Systeme
- Technikgenese und Ko-Evolution
- Computerethische Grundlagen
- Psychologische Grundlagen
- Soziologische Grundlagen
- Digitale Technologien für soziale Dienste
- Digitale Technologien in Organisationen
- Digitale Technologien in der Gesellschaft
- E-Government

Studien-/Prüfungsleistungen:

Abschlusspräsentation

Medienformen:

- Beamer-gestützte Vorlesungen (Folien in elektronischer Form)
- Screencasts und Handouts



Literatur:

- Kienle, A., Kunau, G. (2014). Informatik und Gesellschaft. Eine sozio-technische Perspektive. München: Oldenbourg Wissenschaftsverlag.
- Zweig, K. A., In Neuser, W., In Pipek, V., In Rohde, M., & In Scholtes, I. (2014). Socioinformatics: The social impact of interactions between humans and IT.



Kapitel 26

Screenesign

Lehrform/SWS:

4 SWS: Vorlesung 1 SWS; Seminar/Workshops 3 SWS; Projekt 1 SWS

Arbeitsaufwand:

Gesamtaufwand 150 h, davon

- 40 h Seminar
- 80 h Projektarbeit
- 30 h Selbststudium

Angestrebte Lernergebnisse:

Die Studierenden kennen wesentliche Begriffe der visuellen Kommunikation und können diese anwenden um Briefings, Angebote oder Korrekturwünsche im Design-Kontext zu verstehen oder zu verfassen.

Die Studierenden können Gestaltungslösungen und -kontexte analysieren, argumentieren, diskutieren, dokumentieren und bewerten, um eigene Lösungen innerhalb eines Gestaltungskontextes generieren zu können.



Die Studierenden können in einem gegebenen Gestaltungskontext, unter Berücksichtigung von Gestaltungsregeln (Raster, Layout, Typographie, etc.), eigene Gestaltungslösungen entwickeln, systematisch variieren und argumentieren um gegebene funktionale und/oder kommunikative Ziele zu adressieren.

Inhalt:

Vorlesung:

- Design Basics
- Axis Map & Semantisches Differential
- Kommunikationsmodelle
- Visuelle Wahrnehmung
- Benutzerziele
- Corporate Identity
- Orientierung, Hierarchisierung, Reduktion
- Räumlichkeit
- Gestaltgesetze
- Farbe, Kontraste
- Typographie, Textsatz
- Proportion
- Ordnung, visuelle Struktur, Flow & Transition
- Gestaltungsziele, Gestaltungsprozess

Seminar:

- Designprojekte strukturieren
- Layoutentwicklung mit Wireframes
- Layoutentwicklung für verschiedene Endgeräte



-
- Flow & Transition
 - Typographie & Textsatz
 - Designkonzepte analysieren & bewerten
 - Variantenbildung
 - Modularisierung, Interface Inventar aufbauen & visualisieren

Studien-/Prüfungsleistungen:

Projektpräsentationsprüfung

Medienformen:

Beamergestützte Vorträge, Rechnergestützte Workshops

Literatur:

- Stapelkamp, Torsten: Informationsvisualisierung
- Joachim Böhringer, Peter Bühler & Patrick Schlaich: Kompendium der Mediengestaltung
- Konzeption und Gestaltung für Digital- und Printmedien
- Stapelkamp, Torsten: Screen- und Interfacedesign
- Max Bollwage: Typografie kompakt
- Kerstin Alexander: Kompendium der visuellen Information und Kommunikation
- Maeda, John: Simplicity!: Die zehn Gesetze der Einfachheit
- Lidwell, William; Holden, Kristina; Butler, Jill: Design: Die 100 Prinzipien für erfolgreiche Gestaltung
- Lewandowsky, Pina; Zeischegg, Francis: Visuelles Gestalten mit dem Computer
- Koschembar, Frank: Grafik für Nicht-Grafiker



Kapitel 27

Softwaretechnik

Kurzbeschreibung:

Prinzipien, Methoden und Techniken der modellbasierten methodische objektorientierte Softwareentwicklung

Lehrform/SWS:

4 SWS: Vorlesung 2 SWS; Parktikum 2 SWS

max. 15 Studierende/Praktikumsgruppe;

Arbeitsaufwand:

Gesamtaufwand 150 h, davon

- 36 h Vorlesung
- 36 h Praktikum
- 78 h Selbststudium



Angestrebte Lernergebnisse:

Die Studierenden sollen befähigt werden,

- zu abstrahieren, Modelle zu entwickeln, Unterschiede zwischen Modell und Realität zu beurteilen sowie
- gegebene Modelle zu interpretieren, zu analysieren und zu bewerten,
- komplexe Systeme zu analysieren, im Team zu entwerfen und dabei im Rahmen methodischer Vorgehensweisen Techniken und Werkzeuge der objektorientierten Modellierung und Softwareentwicklung in den Aktivitäten Anforderungsermittlung, Softwarespezifizierung und Entwurf einzusetzen.

Inhalt:

Die Vorlesung gibt zunächst einen Überblick über das Gesamtgebiet Softwaretechnik und behandelt dann ausschließlich grundlegende „Informatikaspekte“ der objektorientierten Softwareentwicklung. Als wesentliche Grundlage werden die wichtigsten Elemente der Unified Modelling Language (UML) vorgestellt und anhand kleinerer Beispiele erläutert. Danach werden typische Aktivitäten der Softwareentwicklung besprochen, wobei die UML als Modellierungssprache benutzt wird. Im Praktikum werden die Anwendung der Modellierungselemente und die Durchführung der Aktivitäten in Gruppenarbeit vertieft.

Das Modul gliedert sich in folgende Inhalte:

- (10%) Softwareentwicklung im Überblick (Komplexität großer Software, Kernaktivitäten und unterstützende Aktivitäten);
- (30%) Die Modellierungssprache UML (Strukturmodellierung mit Objekt- und Klassendiagrammen, Funktionsmodellierung mit Anwendungsfalldiagrammen, Verhaltensmodellierung mit Sequenz-, Kommunikations- und Zustandsdiagrammen);
- (50%) Modellbasierte Softwareentwicklung (Anforderungsermittlung, Softwarespezifizierung und Architekturkonzeption, Entwurfskonzepte und Grobentwurf, Feinentwurf);
- (10%) Zusammenfassung und Ausblick (Modellgetriebene Softwareentwicklung);



Studien-/Prüfungsleistungen:

Klausur 120 Minuten

Medienformen:

- Übungen
- e-Vorlesungen (Video-Clips und Folien in elektronischer Form zum Selbststudium);
- Vertiefende Materialien in elektronischer Form (z.B. SWEBOK)
- Praktika in Kleingruppen, um die erlernten Modelle und Methoden einzuüben und zu vertiefen (Seminarraum, Rechnerlabor); In den Praktika werden Modellierungs- und Entwicklungswerkzeuge eingesetzt.

Literatur:

- Helmut Balzert: Lehrbuch der Software-Technik Bd. I: Basiskonzepte und Requirements Engineering; Spektrum Akademischer Verlag, Heidelberg, 3. Aufl. 2009
- Helmut Balzert: Lehrbuch der Software-Technik Bd. II: Entwurf, Implementierung, Installation und Betrieb; Spektrum Akademischer Verlag, Heidelberg, 3. Aufl. 2012
- Helmut Balzert: Lehrbuch der Software-Technik Bd. III: Software Management; Spektrum Akademischer Verlag, Heidelberg, 2. Aufl. 2008
- Martina Seidl et al.: UML@Classroom; dpunkt.Verlag, Heidelberg, 2012

Unterlagen/Videos: <http://www.uml.ac.at/lernen>

- Winter, M.: Methodische objektorientierte Softwareentwicklung. dpunkt.verlag, Heidelberg, 2005;
- Chris Rupp et al.: UML 2 Glasklar. 4. Aufl., Carl Hanser Verlag, München, 2012
- Jochen Ludewig, Horst Lichter: Software Engineering – Grundlagen, Menschen, Prozesse, Techniken. 2. Aufl., dPunkt Verlag, Heidelberg, 2011



Kapitel 28

Theoretische Informatik 1

Lehrform/SWS:

4 SWS: Vorlesung 2 SWS; Übung 2 SWS

Arbeitsaufwand:

Gesamtaufwand 150 h, davon

- 50h Vorlesung (inkl. Vor- und Nachbearbeitung)
- 50h Übung (inkl. Vor- und Nachbearbeitung)
- 50h Selbstlernphase

Angestrebte Lernergebnisse:

- Grundsätzliches Ziel des Kurses ist eine Einführung in die Begriffe, Methoden, Modelle und Arbeitsweise der Theoretischen Informatik anhand der ausgewählten Teilgebiete.
- Dabei lernen die Studierenden Probleme und Sachverhalte zu abstrahieren und zu modellieren (etwa logische und algebraische Kalküle, graphentheoretische Notationen, formale Sprachen und Automaten sowie spezielle Kalküle wie Petri-Netze)



- Die Studierenden erwerben fundierte Kenntnisse der grundlegenden Themengebiete und eine wesentliche Basis und Vorbereitung für Veranstaltungen in höheren Semestern des Studiums.
- In verschiedenen Grundlagengebieten der Informatik lernen die Studierenden Verfahrenswesen kennen, um den algorithmischen Kern eines Problems zu identifizieren und können passende Algorithmen entwerfen (Automaten, Turing Maschinen, Logik). Dabei können Sie bekannte Problemstellungen im Anwendungskontext erkennen und sind mit den zugehörigen Lösungsmustern vertraut (Modellierung mittels Automaten, Petri-Netzen, Boolescher Algebra, etc.).
- Aufgaben zu den Lehrinhalten (s.u.) werden in kleinen Gruppen (Teamarbeit) selbständig gelöst. Die Lösungen sollen in den Übungsstunden vorgetragen und der Lösungsweg den Kommilitonen hierbei erläutert werden.

Inhalt:

Dieses Grundlagenfach zur Informatik erstreckt sich über zwei Semester (Theoretische Informatik 1 und Theoretische Informatik 2).

- Grundlagen: Mengen, Relationen, Graphen, Polynome; Zahlensysteme, Zahlendarstellung, Numerische Aspekte; Codierung, Informationstheorie
- Logik und Boolesche Algebra: Aussagenlogik; Prädikatenlogik; Boolesche Algebra, Schaltnetze und Schaltwerke
- Reguläre (Typ-3) Sprachen: Endliche Automaten, Reguläre Ausdrücke; Typ3-Grammatiken, Syntaxdiagramme; Chomsky-Hierarchie
- Modellierung sequentieller und paralleler (Ausgabe-) Prozesse: Endliche Maschinen, Berechnungen; Automatenetze, Petri-Netze, Zelluläre Automaten
- Kontextfreie (Typ-2) Sprachen: Kontextfreie Grammatiken, Chomsky- und Greibach-Normalform; Kellerautomaten; Anwendungen (Ableitungs- und Syntaxbäume, Syntax von Programmiersprachen, Backus-Naur-Form)
- Kontextsensitive (Typ-1) und rekursiv aufzählende (Typ-0) Sprachen: Grammatiken, Monotonie, Normalform; Turingautomaten; Einführung in die Begriffe: Berechenbarkeit, Entscheidbarkeit und Komplexität



Studien-/Prüfungsleistungen:

Theoretische Informatik 1 + 2 als eine schriftliche Modulprüfung von 120 Min.

Medienformen:

Vorlesung mit Übung

Literatur:

- Hoffmann, D. (2011): Theoretische Informatik, 2. Auflage
- Brill, M. (2005): Mathematik für Informatiker. Carl Hanser Verlag, München.
- Vossen, G., Witt K. (2004): Grundlagen der Theoretischen Informatik mit Anwendungen. 3. Aufl. Vieweg & Sohn, Braunschweig.
- Hedtstück, U. (2004): Einführung in die Theoretische Informatik. Oldenbourg, München.
- Kelly, J. (2003): Logik. Pearson Studium, München.
- Kelch, R. (2003): Rechnergrundlagen. Von der Binärlogik zum Schaltwerk. Fachbuchverlag Leipzig im Carl Hanser Verlag.
- Dean, N. (2003): Diskrete Mathematik. Pearson Studium. München.
- Asteroth, A., Baier, C. (2002) Theoretische Informatik. Pearson Studium München
- Hopcroft, J. E. et al. (2002): Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie. Pearson Studium, München.
- Meinel, C., Mundhenk, M. (2002): Mathematische Grundlagen der Informatik. B. G. Teubner, Stuttgart.
- Ehrig, H. et al. (1999): Mathematisch-strukturelle Grundlagen der Informatik. Springer, Heidelberg.
- Schöning, U. (1997): Theoretische Informatik - kurzgefaßt. 3. Aufl. Spektrum Akademischer Verlag, Heidelberg.



-
- Schiffmann, W. und Schmitz, R. (1993): Technische Informatik 1. 2.Aufl. Springer, Heidelberg.
 - Urbanski, K. und Voitowitz, R. (1993): Digitaltechnik. BI-Wiss.- Verlag, Mannheim.
 - Beuth, K. (1992): Digitaltechnik. 9.Aufl. Vogel, Würzburg.
 - Morgenstern, B. (1992): Elektronik III, Digitale Schaltungen und Systeme. Vieweg & Sohn, Braunschweig.
 - Rembold, U. et al. (1991): Einführung in die Informatik. 2. Aufl. Hanser, München.
 - Rembold, U. et al. (1990): Aufgaben zur Informatik. Hanser, München.
 - Tietze, U. und Schenk, C. (1990): Halbleiter-Schaltungstechnik. 9.Aufl. Springer, Berlin.



Kapitel 29

Theoretische Informatik 2

Lehrform/SWS:

4 SWS: Vorlesung 2 SWS; Übung 2 SWS

Arbeitsaufwand:

Gesamtaufwand 150 h, davon

- 50h Vorlesung (inkl. Vor- und Nachbearbeitung)
- 50h Übung (inkl. Vor- und Nachbearbeitung)
- 50h Selbstlernphase

Angestrebte Lernergebnisse:

- Grundsätzliches Ziel des Kurses ist eine Einführung in die Begriffe, Methoden, Modelle und Arbeitsweise der Theoretischen Informatik anhand der ausgewählten Teilgebiete.
- Dabei lernen die Studierenden Probleme und Sachverhalte zu abstrahieren und zu modellieren (etwa logische und algebraische Kalküle, graphentheoretische Notationen, formale Sprachen und Automaten sowie spezielle Kalküle wie Petri-Netze)



- Die Studierenden erwerben fundierte Kenntnisse der grundlegenden Themengebiete und eine wesentliche Basis und Vorbereitung für Veranstaltungen in höheren Semestern des Studiums.
- In verschiedenen Grundlagengebieten der Informatik lernen die Studierenden Verfahrenswesen kennen, um den algorithmischen Kern eines Problems zu identifizieren und können passende Algorithmen entwerfen (Automaten, Turing Maschinen, Logik). Dabei können Sie bekannte Problemstellungen im Anwendungskontext erkennen und sind mit den zugehörigen Lösungsmustern vertraut (Modellierung mittels Automaten, Petri-Netzen, Boolescher Algebra, etc.).
- Aufgaben zu den Lehrinhalten (s.u.) werden in kleinen Gruppen (Teamarbeit) selbständig gelöst. Die Lösungen sollen in den Übungsstunden vorgetragen und der Lösungsweg den Kommilitonen hierbei erläutert werden.

Inhalt:

Dieses Grundlagenfach zur Informatik erstreckt sich über zwei Semester (Theoretische Informatik 1 und Theoretische Informatik 2).

- Grundlagen: Mengen, Relationen, Graphen, Polynome; Zahlensysteme, Zahlendarstellung, Numerische Aspekte; Codierung, Informationstheorie
- Logik und Boolesche Algebra: Aussagenlogik; Prädikatenlogik; Boolesche Algebra, Schaltnetze und Schaltwerke
- Reguläre (Typ-3) Sprachen: Endliche Automaten, Reguläre Ausdrücke; Typ3-Grammatiken, Syntaxdiagramme; Chomsky-Hierarchie
- Modellierung sequentieller und paralleler (Ausgabe-) Prozesse: Endliche Maschinen, Berechnungen; Automatenetze, Petri-Netze
- Kontextfreie (Typ-2) Sprachen: Kontextfreie Grammatiken, Chomsky- und Greibach-Normalform; Kellerautomaten; Anwendungen (Ableitungs- und Syntaxbäume, Syntax von Programmiersprachen, Backus-Naur-Form)
- Kontextsensitive (Typ-1) und rekursiv aufzählende (Typ-0) Sprachen: Grammatiken, Monotonie, Normalform; Turingautomaten; Einführung in die Begriffe: Berechenbarkeit, Entscheidbarkeit und Komplexität



Studien-/Prüfungsleistungen:

Theoretische Informatik 1 + 2 als eine Modulprüfung von 120 Min.

Medienformen:

Vorlesung und praktische Übungen

Literatur:

- Hoffmann, D. (2011): Theoretische Informatik, 2. Auflage
- Brill, M. (2005): Mathematik für Informatiker. Carl Hanser Verlag, München.
- Vossen, G., Witt K. (2004): Grundlagen der Theoretischen Informatik mit Anwendungen. 3. Aufl. Vieweg & Sohn, Braunschweig.
- Hedtstück, U. (2004): Einführung in die Theoretische Informatik. Oldenbourg, München.
- Kelly, J. (2003): Logik. Pearson Studium, München.
- Kelch, R. (2003): Rechnergrundlagen. Von der Binärlogik zum Schaltwerk. Fachbuchverlag Leipzig im Carl Hanser Verlag.
- Dean, N. (2003): Diskrete Mathematik. Pearson Studium. München.
- Asteroth, A., Baier, C. (2002) Theoretische Informatik. Pearson Studium München
- Hopcroft, J. E. et al. (2002): Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie. Pearson Studium, München.
- Meinel, C., Mundhenk, M. (2002): Mathematische Grundlagen der Informatik. B. G. Teubner, Stuttgart.
- Ehrig, H. et al. (1999): Mathematisch-strukturelle Grundlagen der Informatik. Springer, Heidelberg.
- Schöning, U. (1997): Theoretische Informatik - kurzgefaßt. 3. Aufl. Spektrum Akademischer Verlag, Heidelberg.



-
- Schiffmann, W. und Schmitz, R. (1993): Technische Informatik 1. 2.Aufl. Springer, Heidelberg.
 - Urbanski, K. und Weitowitz, R. (1993): Digitaltechnik. BI-Wiss.- Verlag, Mannheim.
 - Beuth, K. (1992): Digitaltechnik. 9.Aufl. Vogel, Würzburg.
 - Morgenstern, B. (1992): Elektronik III, Digitale Schaltungen und Systeme. Vieweg & Sohn, Braunschweig.
 - Rembold, U. et al. (1991): Einführung in die Informatik. 2. Aufl. Hanser, München.
 - Rembold, U. et al. (1990): Aufgaben zur Informatik. Hanser, München.
 - Tietze, U. und Schenk, C. (1990): Halbleiter-Schaltungstechnik. 9.Aufl. Springer, Berlin.



Kapitel 30

Visual Computing

Kurzbeschreibung

Das Modul „Visual Computing“ im Medieninformatik Bachelor beschäftigt sich mit der Erzeugung und Verarbeitung visueller Informationen, sowohl in realen als auch computergenerierten Szenarien.

Ziel dieses Moduls ist es den Studierenden eine fachlich fundierte, praktische, sowie theoretische Grundlage im Umgang mit audiovisuellen Medien zu geben. Dabei wird sowohl auf die technische Seite (technischen Grundlagen der Video- und Fernsehtechnik) eingegangen als auch auf die algorithmische (computergenerierte Bildsynthese, Gameentwicklung).

Das Modul ist aus vier Teilbereichen aufgebaut, von denen zwei verpflichtend sind und zwei weitere aus einem Wahlkatalog gewählt werden können.

Die beiden Pflichtkurse schaffen ein Fundament, was es erlaubt innerhalb der beiden verbliebenen Kurse, im Gesamtumfang von 10 CP, tiefer in die jeweilige Spezialisierung einzutauchen. Dabei gibt es grundsätzlich die Möglichkeit sich in Richtung Fernseh- und Videoproduktion oder Gameentwicklung zu vertiefen.

Die Kurse werden nach Verfügbarkeit angeboten.

Die Kurse sind in der Regel projektbasiert aufgebaut, so dass sowohl theoretischer Hintergrund als auch praxisnahes Wissen vermittelt wird und zur Anwendung kommt.

Lehrform/SWS:

Vorlesung, Praktikum / Projekt, Übung mit kursabhängigen Schwerpunkten



Lehrveranstaltungen

Pflichtbereich im Gesamtumfang von 10 CP:

- Audiovisuelle Medientechnik
- Computergrafik und Animation

Aus den folgenden Wahlkatalogen sind zwei weitere Kurse zu jeweils 5CP zu belegen, welche nach Verfügbarkeit angeboten werden.

Wahlkatalog Film/Video:

- Audiovisuelles Medienprojekt 2
- Visuelle Effekte und Animation

Wahlkatalog Game Development:

- Crossplatform Game Development mit Unity3D
- Prozedurale Generierung virtueller Welten

Arbeitsaufwand:

- Audiovisuelle Medientechnik: Gesamtaufwand 150 h, davon - 50h Vorlesung (inkl. Vor- und Nachbearbeitung) - 50h Praktikum / Projekt - 50h Selbstlernphase
- Computergrafik und Animation: Gesamtaufwand 150 h, davon - 50h Vorlesung (inkl. Vor- und Nachbearbeitung) - 50h Praktikum / Projekt - 50h Selbstlernphase
- Audiovisuelles Medienprojekt 2: Gesamtaufwand 150 h, davon - 50h Vorlesung (inkl. Vor- und Nachbearbeitung) - 50h Praktikum / Projekt - 50h Selbstlernphase
- Visuelle Effekte und Animation: Gesamtaufwand 150 h, davon - 50h Vorlesung (inkl. Vor- und Nachbearbeitung) - 50h Praktikum / Projekt - 50h Selbstlernphase
- Crossplatform Game Development mit Unity3D: Gesamtaufwand 150 h, davon - 50h Vorlesung (inkl. Vor- und Nachbearbeitung) - 60h Praktikum / Projekt - 40h Selbstlernphase
- Prozedurale Generierung virtueller Welten: Gesamtaufwand 150 h, davon - 40 h Vorlesung (inkl. Vor- und Nachbereitung), - 40 h Live-Coding und Übungen - 40 h Selbstlernphase - 30 h Begleitetes Abschlussprojekt



Angestrebte Lernergebnisse:

Audiovisuelle Medientechnik

Die Studierenden sollen durch dieses Modul dazu befähigt werden, auf Basis der technischen Grundlagen der Video- und Fernsehtechnik weitergehende Fragestellungen selbstständig zu erarbeiten und sich so auch zukünftige technische Entwicklungen

autonom erschließen zu können.

Neben der Entwicklung und Förderung dieser Fachkompetenz ist die Initiierung der Methodenkompetenz eine wichtige Säule des Vorlesungsmoduls. Unter Methodenkompetenz ist hier die Selbstorganisation im Sinne von wissenschaftlicher Fragestellung an einen Themenkomplex und ein strukturiertes Vorgehen in der Erarbeitung eines Lösungsansatzes zu verstehen. Ziel ist es, das Wissen aus verschiedenen Bereichen, wie Kerninformatik, Internet- und Webtechnologien und benachbarten Wissenschaften mit der in diesem Modul unterrichteten Medientechnologien zu kombinieren und in die Medienproduktion zu integrieren.

Gerade der Umgang mit Technologien und Methoden aus der Film- und Fernsehproduktion erweitert den Erfahrungshorizont der Studierenden über den bekannten Themenbereich der Kerninformatik hinaus und legt ihnen eine Einarbeitung in informatikfremde Sachverhalte und technologische Problemstellungen und deren Lösungsmethoden nahe.

Pragmatisches Ziel ist es, in den unterschiedlichsten Berufsfeldern audiovisueller Medien die Entwicklung und den Einsatz digitaler Medientechnik zu beraten, zu planen, durchzuführen oder zu verantworten.

Computergrafik und Animation

Die Grundlagen der zwei- und insbesondere der dreidimensionalen Computergraphik und Animation stellen ein hervorragendes Paradigma zur Vermittlung zentraler Inhalte und Kompetenzen der Medieninformatik dar.

Den Studierenden wird deutlich, wie der Bogen von den abstrakten, geometrischen und algorithmischen Fakten zu den pragmatischen Gegebenheiten der Computergraphik-Hardware gespannt ist.

Sie erkennen die Zusammenhänge zwischen Grundlagenvorlesungen (Mathematik, Algorithmen, Programmierung) und der Gestaltung von Schnittstellen und Oberflächen und werden so für die jeweiligen Inhalte zusätzlich motiviert.



Dabei lernen Sie im Kontext der Computergrafik Verfahrensweisen um den algorithmischen Kern eines Problems zu identifizieren und können Algorithmen entwerfen, verifizieren und bzgl. ihres Ressourcenbedarfs bewerten.

Sie erwerben die Fähigkeit, aktuelle technologische Entwicklungen im Medieninformatik-Kontext zu bewerten und Trends einzuordnen.

Nach Abschluss des Moduls besitzen die Studierenden grundlegende Kenntnisse über Architektur und Programmierung moderner Graphikhardware, sowie deren Anwendung in konkreten Problemstellungen und Anwendungskontexten.

Am Beispiel von OpenGL und der Rendering-Pipeline lernen die Studierenden Problemstellungen im Anwendungskontext erkennen und sind mit den zugehörigen Lösungsmustern durch praktische Programmierung vertraut.

Das erlernte Wissen und die erlernten Kenntnisse in der Soft- und Grafikhardware-Architektur ermöglicht es erfolgreichen Teilnehmern, anschließend Echtzeit-Visualisierungen mit OpenGL zu implementieren und somit mit einer modernen, plattformunabhängigen API umzugehen, die flexibel an bestehende Anforderungen angepasst werden kann. Zudem haben Sie die Fähigkeit hochparallele Algorithmen auf der Graphikkarte zu entwerfen und auszuführen.

Dabei beherrschen die Studierenden nach Abschluss des Moduls die Fähigkeit abstrakte Szenen- und Objektbeschreibungen zu erstellen und darzustellen, sowie sich in vorhandenen Quelltext einzuarbeiten und diesen sinnvoll weiter zu entwickeln..

Die Inhalte des Moduls befähigen die Studierenden die grundlegenden Algorithmen und Datenstrukturen der Echtzeit-Computergraphik zu beherrschen.

Die Studierenden können ihr erworbenes Können und Wissen zur Implementierung einer eigenen Game/Visualisierungs-Engine einsetzen. Dies zeigen Sie durch Umsetzung eines eigenen Projektes in Kleingruppen, wo sie zusätzlich lernen mündlich überzeugend zu präsentieren, abweichende Positionen zu erkennen und in eine sach- und interessengerechte Lösung zu integrieren. Sie zeigen dadurch, dass Sie in der Lage sind sich selbstständig neues Wissen anzueignen und zu erkennen, welches Wissen relevant ist, können mediengestalterische Grundkompetenzen anwenden und besitzen aktive Vokabularen zur Beschreibung und Realisierung angemessener Konzeptionen. Zudem können sie die Realisationen bezüglich der Zielsetzungen kritisch diskutieren.

Audiovisuelles Medienprojekt 2

Die praktische Umsetzung des Vorlesungsstoffes, die Kommunikation und Zusammenarbeit im Team über Themenbereiche dieses Faches und die Präsentation von eigenen Projekten und Untersuchungsergebnissen sind die Lernziele des Moduls Audiovisuelles Medienprojekt 2. Neben dieser formulierten Fachkompetenz, Methodenkompetenz und Kommunikationskompetenz ste-



hen gerade die sogenannten Softskills Teamfähigkeit und Kommunikationsfähigkeit im Focus der Ausbildung in diesem Modul.

Die Studierenden kennen über die grundlegenden Erzählformen audiovisueller Medien hinaus spezielle Formate wie Spielfilm, Imagefilm und Studioproduktion und haben dabei folgende Fertigkeiten: Sie können eigene audiovisuelle Erzählformen auf der Basis dieser Erzählmuster entwickeln und sind befähigt zur Analyse, zur Diskussion und zur kritischen Betrachtung audiovisueller Medieninhalte.

Pragmatisches Ziel ist es, in den unterschiedlichsten Berufsfeldern digitaler audiovisueller Medien die Entwicklung und den Einsatz audiovisuellen Content zu beraten, zu planen, durchzuführen oder zu verantworten.

Visuelle Effekte und Animation

Die Studierenden kennen die grundlegenden Produktionsschritte und Abläufe einer Film- und TV-Produktion mit visuellen Effekten sowie die in diesem Zusammenhang eingesetzten Softwaretools.

Sie haben die Fertigkeit, spezifische Fragestellungen der Umsetzung visueller, computerbasierter Effekte und der damit zusammenhängenden Bildbearbeitung zu bearbeiten und fallbezogene individuelle Lösungen zu entwickeln.

Unter Entwicklungs- und Methodenkompetenz auf dem Gebiet der Visual Effects ist die Fähigkeit zu verstehen, eigene und für den jeweiligen Anwendungsfall auch eventuell neue Lösungsansätze zu entwickeln, bei denen die unterschiedlichen Methoden der Visual Effects-Ausführung und Bearbeitung zum Einsatz kommen. Nachdem die Planung, Durchführung und die Bearbeitung von Projekten auf dem Gebiet der Film- und TV-Produktion mit visuellen Effekten in der Regel im kleinen Team erfolgt sind gerade die Softskills der Teamkompetenz und der Organisationskompetenz von großer Wichtigkeit in diesem Modul.

Berufsbilder, die von diesem Modul angesprochen werden, sind zum einen in der Visual-Effects-spezifischen Softwareentwicklung, als auch im Anwendungskontext zu finden: So zum Beispiel in der Planung, Organisation, Durchführung und Verantwortung von VFX-Projekten.

Cross-Platform Game Development mit Unity 3D

Die Studierenden kennen wesentliche Konzepte und Technologien des Game Developments mit Unity 3D und können diese anwenden, um eigenständig im Team Interaktive Applikationen zu konzipieren, realisieren und optimieren.

Die Studierenden kennen die grundlegenden Möglichkeiten von Game Engine Frameworks und sind in der Lage diese kritisch zu beurteilen und auf Basis der Anforderungen eines konkreten



Projekts die Umsetzungsmöglichkeiten und Vorgehen zu evaluieren und entsprechende Strategien zu entwickeln, sowie kritisch die benötigten Bibliotheken und Komponenten auszuwählen und diese Wahl zu begründen.

Die Kursteilnehmer sammeln im Rahmen ihres eigenständig entwickelten Projektes Erfahrungen in der Entwicklung von Kleinprojekten bis mindestens zum Grad einer spielbaren Alphaversion oder eines Prototypen.

Prozedurale Generierung virtueller Welten

Die Studierenden haben die Möglichkeit ihr Wissen über 3D-Computergrafik, 3D-Geometrie und Programmierung zu vertiefen und praktisch anzuwenden. Durch den Einsatz von Unity als Crossplatform-Game-Development-Tool können die Studierenden ihre Erfahrung mit einer aktuellen Game-Engine und der dazugehörigen Entwicklungsumgebung vertiefen und werden befähigt diese um Funktionalitäten zu erweitern.

Die Inhalte des Moduls befähigen die Studierenden die grundlegenden Algorithmen und Datenstrukturen der Echtzeit-Computergrafik mit der Game-Engine Unity zu beherrschen.

Das Modul ist daher geeignet das Wissen aus den Modulen „Computergrafik und Animation“, sowie „Cross-Platform Game Development mit Unity 3D“ zu vertiefen und zu erweitern.

Konzepte aus der 3D-Computergrafik und der prozeduralen Generierung werden in diesem sehr praktisch ausgelegten Kurs implementiert und die relevanten Grundlagen vermittelt, die es den Studierenden ermöglichen eigenständig Anforderungen für Projekte mit prozeduralen Techniken zu entwickeln und diese praktisch umzusetzen und kritisch zu evaluieren.

Nach Abschluss des Moduls besitzen die Studierenden grundlegende Kenntnisse über verschiedene Techniken der prozeduralen Generierung zur Erstellung von 3D Content für Visualisierungen, Simulationen und Spiele, sowie die Fähigkeiten diese einzusetzen, sowie selbst zu entwickeln.

Je nach gewählten Vertiefungskursen entwickeln die Studierenden Fähigkeiten zur selbstverantwortlichen Durchführung von Projekten im Bereich Gamedevelopment, dreidimensionaler Darstellung virtueller Szenen, Film- und Fernsehtechnik, sowie Visueller Effekte.

Im Detail:



Inhalt:

Audiovisuelle Medientechnik

- Grundlagen der Fernsehtechnik
- Digitale Fernsehtechnik
- HD-Technik
- Videodatenreduktion
- Bildwandler
- Das Optische System der Videokamera
- Signalverarbeitung in der Videokamera
- Signalaufzeichnung
- Elektroakustik
- Bildwiedergabesysteme
- Lichttechnik und Beleuchtung

Computergrafik und Animation

- Graphikhardware,
- OpenGL
- Transformationen und homogene Koordinaten
- Interpolation
- Kameramodelle
- Clipping
- Shaderprogrammierung
- Animation
- Texturierung



-
- Fortgeschrittene Algorithmen (Schatten, Reflexionen, Bump-, Normal-, Parallax-, Relief-Mapping), Globale Beleuchtung, Deferred Shading)
 - Perzeption
 - Grundlagen des Ray Tracings

Audiovisuelles Medienprojekt 2

- Vertiefung der Video- und Audioaufnahmetechnik
- Verschiedene Dramaturgiemodelle
- Drehbuch, Auflösung, Storyboard
- Schnitt und Montage
- Liveproduktion im Studio
- Medienproduktion in den Formaten Spielfilm, Imagefilm und Studioproduktion

Visuelle Effekte und Animation

- Storyboard
- Kalkulation
- Produktionabläufe
- Keyverfahren mit Green- und Bluescreen
- Compositing
- Umgang mit Bild-/Videobearbeitungswerkzeugen

Cross-Platform Game Development mit Unity 3D

- Aufbau einer Game Engine
- Gameobjects
- Game Physics
- Interaktion



- Spielmechaniken

Prozedurale Generierung virtueller Welten

- Einführung in die Game-Engine Unity
- Grundlagen 3D-Geometrie
- Erweiterung des Unity-Editors
- Primitive und Mesh-Datenstrukturen
- UV-Mapping und Texturierungstechniken/Materialien
- Prozedurale Texturgenerierung
- Parametrisierung von 3D-Modellen
- Kurven und Flächen
- Height-Maps
- L-Systeme und „Turtle“-Grafik-Renderer
- Triangulations-Algorithmen für Polygone
- Voxel-Terrain-Generierung

Studien-/Prüfungsleistungen:

Audiovisuelle Medientechnik

Schriftliche Prüfung von 90 Min

Computergrafik und Animation

Die erfolgreiche Teilnahme an den Praktikas ist Voraussetzung für die Klausur (120 Minuten) oder mündliche Prüfung



Audiovisuelles Medienprojekt 2

Projektarbeit und schriftliche Ausarbeitung

Visuelle Effekte und Animation

Projektarbeit und schriftliche Ausarbeitung

Cross-Platform Game Development mit Unity 3D

Präsentation und Dokumentation eines eigenständig entwickelten Projekts

Prozedurale Generierung virtueller Welten

Die erfolgreiche Teilnahme am Abschlussprojekt (eigenständiges Projekt, auch in Kleingruppen möglich) und Fachgespräch

Teilprüfungen in den jeweiligen Kursen.

Medienformen:

Beamergestützte Vorträge, Rechnergestützte Workshops

Literatur:

Audiovisuelle Medientechnik

- Schmidt Ulrich, Professionelle Videotechnik, Springer-Verlag Berlin Heidelberg New York 2013, ISBN 978-3-642-38992-4
- Johannes Webers, Film- und Fernsehtechnik, Franzis Verlag, Pöng 2000, ISBN 3-7723-7116-7
- Möllering, Slansky, Handbuch der professionellen Videoaufnahme Edition Filmwerkstatt, Essen, 1993, ISBN 3 - 9 802 581 - 3 - 0



Computergrafik und Animation

- Peter Shirley, Fundamentals of Computer Graphics, Peters, Wellesley
- Andrew Woo, et al., OpenGL Programming Guide, Version 4.3, Addison-Wesley,
- Tomas Akenine-Möller, Eric Haines, und Naty Hoffman, Real-Time Rendering, 3. Ausgabe, Peters, Wellesley
- Randi J. Rost, John M. Kessenich, Barthold Lichtenbelt, OpenGL Shading Language, 2. Ausgabe, Addison-Wesley
- Alan Watt, 3D Computer Graphics, Addison-Wesley
- Frank Nielsen, Visual Computing, Charles River Media, 2005
- James Foley, Andries Van Dam, et al., Computer Graphics : Principles and Practice, 2. Ausgabe, Addison-Wesley

Audiovisuelles Medienprojekt 2

- James Monaco, Film verstehen, Rowolth Taschenbuch Verlag Hamburg, 1980, ISBN 3-499-162717
- Syd Field, Drehbuchschreiben für Film und Fernsehen, München 2003, ISBN 354836473X
- Steven D. Katz, Die Richtige Einstellung, Zweitausendeins, Frankfurt a.M.1998,ISBN 3-86150-229-1
- David Lewis Yewdall, Practical Art of Motion Picture Sound, Focal Press, USA 2003, ISBN 0-240-80525-9
- Hans Kornacher & Manfred Stross, Dokumentarisches Videofilmen, Augustus Verlag, Augsburg, 1992, ISBN 3-8043-5474-2
- Hans Beller Hg., Handbuch der Filmmontage, München: TR-Verlagsunion, 1993, ISBN 3-8058-2357-6
- Karel Reisz, Gavin Millar, Geschichte und Technik der Filmmontage, München: Filmlandpresse, 1988, ISBN 3-88690-071-1
- Chris Vogler, Die Reise des Drehbuchschreibens, Verlag Zweitausendeins
- Wolfgang Lanzenberger, Michael Müller, Unternehmensfilme drehen: Business Movies im digitalen Zeitalter, ISBN 978-386764367



Visuelle Effekte und Animation

- Flückiger Barbara, Visual Effects: Filmbilder aus dem Computer (Zürcher Filmstudien), Schüren Verlag GmbH, 2008, ISBN 978-3894725181
- Bertram Sascha, VFX (Praxis Film), UVK, 2005, ISBN 978-3896695154

Cross-Platform Game Development mit Unity 3D

- Unity 3D API (<https://docs.unity3d.com/ScriptReference/>)
- Unity 3D Manual (<https://docs.unity3d.com/Manual/index.html>)
- Unity Tutorials (<https://unity3d.com/de/learn/tutorials>)
- Ian Millington, John Funge, Artificial Intelligence For Games, Second Edition, CRC Press, 2009
- Mat Buckland, Programming Game AI by Example, Wordware Game Developers Library, 2004
- Steve Rabin et al., AI Game Programming Wisdom 1-4, Cengage Learning

Prozedurale Generierung virtueller Welten

- Peter Shirley, Steve Marschner, "Fundamentals of Computer Graphics", CRC Press
- David Salomon, "Curves and Surfaces for Computer Graphics", Springer
- Carsten Seifert, "Spiele entwickeln mit Unity 5", Hanser
- Noor Shaker, Julian Togelius, Mark J. Nelson, "Procedural Content Generation in Games (Computational Synthesis and Creative Systems)", Springer
- Ryan Watkins, "Procedural Content Generation for Unity Game Development", Packt Publishing
- Dale Green, "Procedural Content Generation for C++ Game Development", Packt Publishing



Kapitel 31

Web Development

Kurzbeschreibung

Einführungen in Konzepte, Techniken und Arbeitsweisen der Web Entwicklung.

Lehrform/SWS:

6 SWS Vorlesung, 6 SWS Praktikum, 4 SWS Projekt

Lehrveranstaltungen

- Web Frontend Entwicklung
- Serverseitige Frameworks und Dienste
- Internet of Things
- Web Development Projekt

Arbeitsaufwand:

- Web Frontend Entwicklung: Vorlesung, Seminar(50 Stunden); Selbstlernphase (100 Stunden)



- Serverseitige Frameworks und Dienste: Vorlesung, Seminar(50 Stunden); Selbstlernphase (100 Stunden)
- Internet of Things: Vorlesung, Seminar (50 Stunden); Selbstlernphase (100 Stunden)
- Web Development Projekt: Projekt(150 Stunden)

Angestrebte Lernergebnisse:

Web Frontend Entwicklung:

Die Studierenden kennen wesentliche Konzepte und Technologien des Web-Frontend Developments und können diese anwenden, um eigenständig im Team Web-Frontends zu konzipieren, realisieren und optimieren.

Die Studierenden sind in der Lage ein gegebenes Gestaltungskonzept zu verstehen und zu erweitern, um dies als Web-Frontend umzusetzen.

Die Studierenden kennen Web-Frontend Frameworks und sind in der Lage diese kritisch zu beurteilen und auf Basis der Anforderungen eines konkreten Projekts das optimale Framework Set zu konfektionieren und die Auswahl zu begründen.

Die Studierenden kennen das Zusammenspiel von server- und clientseitigen Komponenten im Bereich des Webs und können Web-Frontends konzipieren und realisieren, die mit serverseitigen Komponenten und Diensten möglichst optimal zusammen arbeiten. Sie können außerdem, bezogen auf eine konkrete Aufgabenstellung, abwägen, welche Funktionalitäten clientseitig und welche serverseitig gelöst werden sollten.

Serverseitige Frameworks und Dienste:

Die Studentinnen und Studenten kennen

- wesentliche Frameworks, Dienste und Werkzeuge für die serverseitige Entwicklung von Web Anwendungen
- können ausgewählte Frameworks, Dienste und Tools in einem Projektkontext anwenden.

Die Kompetenz zur systematischen Entwicklung von Systemen in einem arbeitsteiligen Team wird eingeübt und vertieft. Kenntnisse aus den anderen Modulen der Vertiefung werden vertieft und verknüpft und im Rahmen eines konkreten Projektauftrags angewendet.



Die Studierenden sind in der Lage ein Projektbriefing zu durchdringen und daraus einen Projektauftrag abzuleiten und diesen im Team abzuarbeiten.

Den Teilnehmern steht eine Auswahl an Techniken und Frameworks zur Verfügung, aus dem sie die passenden Ansätze begründet auswählen und anwenden können.

Die StudentInnen sind in der Lage eine komplexe Anwendung im Web über mehrere Endgeräte hinweg zu planen, zu realisieren und zu dokumentieren.

Internet of Things:

In diesem Modul lernen die Teilnehmer das Gebiet Internet of Things kennen. Dabei liegt ein besonderer Fokus auf der Bedeutung des Web für Applikationen jenseits eines Browsers. Immer mehr Alltagsgegenstände werden mit Technologien angereichert, die eine Dienste-Bereitstellung oder Dienst-Nutzung über das Web ermöglichen (beispielsweise das Steuern von Gegenständen oder das Erfassen von Sensordaten). In diesem Modul werden relevante Konzepte und aktuelle Technologien für das Internet der Dinge diskutiert und in prototypischen Anwendungen erprobt.

Studierende können nach diesem Modul selbstständig Anwendungen für das Internet of Things konzipieren und realisieren, indem Sie

- hardware-nahe Aspekte im Design der Applikation berücksichtigen,
- eine System-Architektur entwerfen,
- kriterien-basiert geeignete Technologien zur Realisierung auswählen,
- eine prototypische Anwendung implementieren,
- inkrementell-iterativ vorgehen und Projektentscheidungen auf der Grundlage vorliegender Zwischenstände treffen.

Die Studierenden

- kennen ausgewählte Methoden und Frameworks für die Web Entwicklung im Front-End, im Back-End und in vernetzten Geräten (IoT),
- können eine Methoden und Technologiewahl für einen Projektkontext fachlich begründen,
- können Frameworks und Methoden zur Realisierung von Proof-of-Concepts in einem Projektkontext einsetzen und
- können die erzielten Ergebnisse fachlich, kritisch einordnen und diskutieren,

um kompetent in Web Entwicklungs Teams mitwirken zu können.



Inhalt:

Web Frontend Entwicklung:

- Web Basics: HTML, CSS, Javascript
- CSS: Komplexe Layouts & Responsivität
- Javascript: Dynamische Anwendungen
- Media Types
- CSS Frameworks
- CSS Preprozessoren
- Javascript Frameworks
- Performance
- Microdata, Internationalisierung, SEO, Barrierefreiheit

Serverseitige Frameworks und Dienste:

- NodeJS
- Services im Web: Amazon WS (AWS), Google Firebase,
- NoSQL Datenbanken
- Web Analyse: Piwik,

Ausgewählte Tools sollen tiefgreifend erarbeitet werden und in einem Projektkontext angewendet werden. Dies erfolgt in der Regel in dem begleitenden Projekt

Internet of Things:

Zu Beginn des Moduls findet eine Einführung durch den Dozenten statt. Danach werden in seminaristischem Unterricht verschiedene aktuelle und relevante Technologie des Internet of Things vermittelt, demonstriert und erprobt (bspw. Raspberry Pi, Arduino, MQTT, Node-Red, Johnny Five, AndroidThings). Diese werden in Teams zur Entwicklung eines eigenen IoT-Prototypen eingesetzt.

Weitere Inhalte sind:



-
- Physical Computing
 - Prototyping und Retrofitting
 - Hardware (bspw. RaspberryPi und Arduino)
 - Sensoren und Aktoren
 - Frameworks (bspw. NodeRed und Johnny Five)
 - Architekturen und Protokolle (bspw. event-basierte Architekturen und MQTT)
 - Mobile Web und Smartphone Sensoren (bspw. GPS, Beacons)

Studien-/Prüfungsleistungen:

Projektarbeit mit Projektpräsentationsprüfung und Fachgespräch.

Medienformen:

Beamergestützte Vorträge, Rechnergestützte Workshops

Literatur:

- Randy Connolly, Ricardo Hoar: Fundamentals of Web Development
- Andy Clark: Hardboiled Web-Design
- Tilkov et al.: REST und HTTP- Entwicklung und Integration nach dem Architekturstil des Web, dpunkt.verlag 2015
- Watkin: Practical XMPP, Packt Publishing 2016
- Saint-Andre: XMPP: The Definitive Guide, O'Reilly 2009
- Roy: RabbitMQ in Depth, Manning 2016
- Newman: Building Microservices: Designing fine-grained systems, O'Reilly 2015



Kapitel 32

Social Computing

Kurzbeschreibung

In der Vertiefung „Social Computing“ werden die Wechselwirkungen zwischen Gesellschaft und Informatik in den Mittelpunkt gestellt. Rechnersysteme und Netzwerke werden von Menschen intentional gestaltet, ausgerichtet an gesellschaftlichen Normen, Prozessen und Bedürfnissen. Gleichzeitig beeinflussen IT-Systeme diese gesellschaftlichen Normen und verändern Prozesse in allen Lebensbereichen. Die verantwortungsbewusste Konzeption und Realisierung von soziotechnischen Systemen (z.B. Social Software, Online Communities, e-Health, e-Government und e-Learning Angebote) sowie die empirische Evaluation existierender Systeme sind zentrale Ziele. Lösungen sollen unter ganzheitlichen Gesichtspunkten entwickelt werden. Verschiedene Wertvorstellungen und Interessen unterschiedlicher Stakeholder müssen identifiziert und berücksichtigt werden.

Der Schwerpunkt verbindet daher Theorien, Modelle und Methodik der Human- und Sozialwissenschaften mit anwendungsorientierter Informatik. Studierende sollen in der Lage sein, computergestützte Systeme nach ethischen, politischen, sozialen und psychologischen Kriterien zu bewerten, zu planen und umsetzen zu können.

Ziel ist es, soziale Innovation durch digitale Anwendungen entstehen zu lassen. Neben den empirischen Methoden werden Designmethoden vermittelt, sowohl auf der konzeptionellen als auch auf der softwaretechnischen Implementierungsebene, um robuste, sichere und flexible Systeme zu gestalten.

Das Modul ist aus vier Teilbereichen aufgebaut, von denen zwei verpflichtend sind und zwei weitere aus einem Wahlkatalog gewählt werden können.

Verpflichtende Kurse im Umfang von je 5 CP sind:

- Soziotechnische Systeme



- Empirische Forschungsmethoden
- Gamification
- Projekt

Die Kurse sind in der Regel projektbasiert aufgebaut, so dass sowohl theoretischer Hintergrund als auch praxisnahes Wissen vermittelt wird und zur Anwendung kommt.

Lehrform/SWS:

Vorlesung, Praktikum / Projekt, Übung mit kursabhängigen Schwerpunkten

Arbeitsaufwand:

Gesamtaufwand 600 h, davon

- 150h Soziotechnische Systeme
- 150h Empirische Forschungsmethoden
- 150h Gamification
- 150h Projekt

Angestrebte Lernergebnisse:

siehe jeweilige Fächerbeschreibung

Inhalt:

siehe Modulbeschreibungen:

- Soziotechnische Systeme
- Empirische Forschungsmethoden
- Gamification



-
- Projekt

Studien-/Prüfungsleistungen:

Teilprüfungen in den jeweiligen Kursen.

Medienformen:

siehe jeweilige Kursbeschreibung

Literatur:

siehe jeweilige Kursbeschreibung



Kapitel 33

Wahlpflichtmodul

Lehrform/SWS:

Je nach Modul

Arbeitsaufwand:

150 Stunden

Angestrebte Lernergebnisse:

Fachliche Vertiefung oder Verbreiterung, nach persönlichem Interesse. Es kann eines der Module aus dem Katalog aller Module der Informatik Bachelorstudiengänge gewählt werden. Auch Pflichtmodule anderer Informatik Studiengänge am Campus können als Wahlpflichtmodule in der Medieninformatik belegt werden.

Inhalt:

Je nach Modul



Studien-/Prüfungsleistungen:

Je nach Modul

Medienformen:

Je nach Modul

Literatur:

Je nach Modul