

Modulhandbuch für den Studiengang Bachelor Informatik

Begriffserläuterungen	3
1. Studienabschnitt	4
Theoretische Grundlagen	4
Grundlagen der Informatik	5
Mathematik I (Algebra)	7
Mathematik II (Analysis)	8
Theoretische Informatik	9
Praktische Informatik	10
Programmieren I	11
Programmieren II	13
Software Engineering	14
Grundlagen der Betriebswirtschaftslehre	15
Grundlagen der Betriebswirtschaftslehre	16
Englisch	16
Englisch I/II	17
Englisch III/IV (IN)	18
2. Studienabschnitt	18
allgemeines Modul	18
Allgemeinwissenschaftliche Wahlpflichtmodule	19
Theoretische Grundlagen	19
Algorithmen und Datenstrukturen	20
Kryptographie und Informationssicherheit	22
Mathematik III (Statistik)	24
Softwaremethoden und -technologien	24
Datenbanken	25
Interaktive Systeme	27
IT-Anwendungen (Bachelor IN)	28
Programmiersprachen	29
Softwarearchitektur	30
Web-Programmierung	32
Systeminfrastrukturen	33
Betriebssysteme	34
Rechnerkommunikation und -netze	35
Rechnersysteme	36
Praktisches Studiensemester	37
Praktikum	38
Praxisseminar	39
Projektmanagement	40
Fachwissenschaftliche Wahlpflichtfächer	41
Schwerpunkt-FWPF (min. 16SWS/20LP, max. SWS/LP)	41
Aktuelle Trends im Einsatz von Projektmanagement-Methoden	42
Algorithmische Graphentheorie	43
Autonome mobile Roboter	44
Betriebssysteme für einfache Mikrocontroller	46
Clean Code in Java-Projekten	48
Datenschutz	50
Einführung in die Funktionale Programmierung	52
Einführung in Operations-Research	54
Embedded Systems	56
Graphical User Interface Design and Information Visualization	58
Grundlagen Content-Management-Systeme	59
Grundlagen der Computerspielentwicklung	61
Grundlagen der Software Security	63
Grundlagen des maschinellen Lernens	65
Grundlagen des Softwaretests	66
Grundlagen von C++ für C#-Entwickler	68

In-Memory Computing am Beispiel von SAP HANA	70
Lean Startup	72
Logik	74
Praxis des Software-Tests eingebetteter Systeme	75
Programmieren von Mikrocontrollern	77
Programmierung und Technologie betrieblicher Standardsoftware (SAP-Praktikum)	79
Requirements Engineering	80
Software-Ergonomie in der Medizintechnik	82
Software-Zuverlässigkeit: Theorie und Praxis	83
Technical Computing	84
Text Analytics	85
Virtuelle Maschinen	87
Visualisierung	89
Voice-over-IP	91
Web Application Security	93
wählbar außerhalb Schwerpunkt	94
Datenbanken II in der Wirtschaftsinformatik	95
Digitalisierung und Nachhaltigkeit	97
Einführung in Business Intelligence	100
Einsatz von Simulations-Software	102
Electronic Commerce	103
Finanzmathematik	105
IT- und Wirtschaftsrecht	106
Spieltheorie	107
Technologien und betriebswirtschaftliche Bedeutung von Industrie 4.0	109
Abschlussarbeit	110
Bachelorarbeit	111
Seminar zur Bachelorarbeit	112
Hilfsmittel	113

Begriffserläuterungen

ECTS	<p>European Credit Transfer System: Diese Vereinbarungen zur Anrechnung, Übertragung und Akkumulierung von Studienleistungen basieren auf dem Arbeitspensum, das Studierende durchzuführen haben, um die Ziele des Lernprogramms zu erreichen. Für jede studienbezogene Leistung wird der voraussichtliche durchschnittliche Arbeitsaufwand angesetzt und auf das Studienvolumen angerechnet. Der Arbeitsaufwand umfasst Präsenzzeit und Selbststudium ebenso wie die Zeit für die Prüfungsleistungen, die notwendig sind, um die Ziele des vorher definierten Lernprogramms zu erreichen. Mit dem ECTS können Studienleistungen international angerechnet und übertragen werden.</p>
Arbeitsaufwand (Workload) und Leistungspunkte (ECTS-LP)	<p>Der Arbeitsaufwand der Studierenden wird im ECTS in credits (credit points) angegeben.</p> <p>Deutsche Übersetzungen für credit point sind die Begriffe Leistungspunkt oder ECTS-Punkt. Ein Arbeitsaufwand von 30 Zeitstunden bedeutet einen Leistungspunkt.</p> <p>Der Arbeitsaufwand von Vollzeitstudierenden entspricht 60 Leistungspunkten pro Studienjahr, also 30 Leistungspunkten pro Semester. Das sind 1.800 Stunden pro Jahr oder 45 Wochen/Jahr mit 40 Stunden/Woche.</p> <p>Der Arbeitsaufwand setzt sich aus verschiedenen Bestandteilen zusammen:</p> <ul style="list-style-type: none">• Präsenzzeit,• Zeit für die Vor- und Nachbereitung des Vorlesungsstoffs,• Zeit für die Vorbereitung von Vorträgen und Präsentationen,• Zeit für die Erstellung eines Projekts,• Zeit für die Ausarbeitung einer Studienarbeit,• Zeit für notwendiges Selbststudium,• Zeit für die Vorbereitung auf mündliche oder schriftliche Prüfungen. <p>Die siebensemestrigen Bachelorstudiengänge bescheinigen erfolgreichen Studierenden also 210 ECTS-LP, die dreisemestrigen Masterstudiengänge weitere 90 ECTS-LP. Damit ist die Forderung nach 300 ECTS-LP für ein erfolgreich abgeschlossenes Masterstudium erfüllt.</p>
Semesterwochenstunden und Präsenzzeit	<p>Eine Semesterwochenstunde ist die periodisch wiederkehrende Lehrinheit in einem Modul, in der Regel im Rhythmus von einer oder zwei Wochen. Dabei wird eine Präsenz von 45 Minuten plus Wegzeiten gerechnet, sodass die Vorlesungsstunde als eine Zeitstunde gewertet wird.</p> <p>Wir rechnen mit einer Vorlesungszeit von 16 Wochen pro Semester, wodurch sich aus der Zahl der Semesterwochenstunden die geforderte Präsenzzeit („Kontaktzeit“) direkt ableitet:</p> <p>1 SWS entspricht 16 Stunden Präsenzzeit. Natürlich wird gerundet.</p>
Module	<p>Das Studium ist inhaltlich in Module aufgeteilt, die zur besseren Übersicht in Modulgruppen zusammengefasst sind.</p> <p>Jedes Modul wird mit einer Prüfung abgeschlossen und ist mit Leistungspunkten versehen, die dem Studierenden bei erfolgreichem Ablegen der Prüfung gutgeschrieben werden.</p>
Studienbegleitende Prüfungen	<p>Sämtliche Prüfungen erfolgen über das gesamte Studium verteilt studienbegleitend und stehen in direktem Bezug zur Lehrveranstaltung. Prüfungsbestandteile können je nach Lehrveranstaltung veranstaltungsbegleitend oder nach Abschluss des Moduls stattfinden, beispielsweise als Referat, Klausurarbeit, mündliche Prüfung, Hausarbeit mit Kolloquium, Entwurf mit Kolloquium, Laborbericht, Exkursionsbericht oder einer Kombination. In den Beschreibungen der einzelnen Module wird im Modulhandbuch die jeweilige Prüfungsform festgelegt.</p> <p>Gemäß der Studien- und Prüfungsordnung kann Fachwissenschaftliche</p>

Wahlpflichtmodule nur ableisten,
wer alle Module des ersten Studienabschnitts bestanden hat.
Einzelheiten zur Organisation der Fachwissenschaftlichen Wahlpflichtmodule regelt
der Studienplan.

Liste Hilfsmittel

- 1) keine Hilfsmittel
- 2) unbeschränkt
- 3) Vorlesungsmitschrift
- 4) Vorlesungsumdrucke
- 5) selbstgefertigte Arbeitsunterlagen (DinA4 Seitenzahl; seit selbstgefertigte
Arbeitsunterlagen müssen geheftet und mit dem Namen des/der Studierenden
versehen sein)
- 6) mathematische Formelsammlung
- 7) Taschenrechner, nicht programmierbar (Taschenrechner mit vollständiger
alphanumerischer Tastatur und/oder Graphikdisplay sind nicht erlaubt)
- 8) Gesetzestexte, z.B. BGB, UrhG, PatG, UWG, Betr.VG, BDSG, StGB, Stopp, TKG
- 9) Lehrbuch

Grundlagen der Informatik

Studiengang	Bachelor Informatik 1. Studienabschnitt Theoretische Grundlagen
Modul	Grundlagen der Informatik
Modulverantwortliche	Rainer Weber, Thomas Fuhr, Jörg Roth, Florian Gallwitz, Michael Zapf
Dozent(en)	Rainer Weber, Michael Zapf
Vorkenntnisse	Schulwissen
Arbeitsaufwand	210 Stunden, davon: 95 Stunden Präsenzzeit 95 Stunden Vor- und Nachbereitung des Lehrstoffs 20 Stunden für Prüfung und Prüfungsvorbereitung
Leistungspunkte	7
Semesterwochenstunden	6
Veranstaltungstyp	4 SWS Vorlesung mit 2 SWS Übung
Semesterturnus	Wintersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Schaffen von Grundlagen, insbesondere im Hinblick auf technische Abläufe in einem Computersystem sowie auf die Repräsentation von Informationen
Lernziel	Kenntnis und Verständnis der Repräsentation von Daten wie ganze und reelle Zahlen und Zeichenketten; Verständnis der mathematischen Grundlagen von Rechenoperationen auf diesen Repräsentationen und selbständige Anwendung; Befähigung zur Konvertierung zwischen Repräsentationen mit besonderem Gewicht auf dem Binär-, Oktal- und Hexadezimalsystem. Analyse von logischen Schaltungen; Verständnis der mathematisch-logischen Grundlagen; selbständige Gestaltung von Logikschaltungen auf Basis eines vorgegebenen Verhaltens; Verständnis für den grundlegenden Entwurf von Computersystemen; Analyse und Erstellung von Maschinensprache- und Assemblerprogrammen.
Schlüsselqualifikation	Fähigkeit zu abstraktem mathematischen Denken und Schließen, technisches Verständnis
Lehrinhalte	<ul style="list-style-type: none">• Repräsentation positiver ganzer Zahlen in verschiedenen Stellenwertsystemen• Negative ganze Zahlen in verschiedenen Stellenwertsystemen• Reelle Zahlen im Standardformat (IEEE-754)• Rechenoperationen und Konvertierungen• Boolesche Algebren• Boolesche Ausdrücke und Funktionen• Schaltnetze und Schaltwerke• Allgemeine Computerarchitektur• Maschinensprache am Beispiel der MIPS-Architektur• Assemblersprache der MIPS-Architektur
Literatur	<ul style="list-style-type: none">• R. Klar: Digitale Rechenautomaten, Berlin, 1988, Walter de Gruyter.

- W. Oberschelp, G. Vossen: Rechneraufbau und Rechnerstrukturen, 9. Auflage, München, 2003, Oldenbourg Verlag.
- Y. Patt, S. Patel: Introduction to Computing Systems - From Bits to Gates and Beyond, 2nd ed., 2004, McGraw Hill.
- D. A. Patterson, J. L. Hennessy: Computer Organization & Design, 2nd ed., San Francisco, CA, 1998, Morgan Kaufman Publishers Inc.
- A. S. Tanenbaum: Computerarchitektur. Strukturen - Konzepte - Grundlagen. 2005. Pearson Studium.
- H. Delfs, H. Knebl, C. Schiedermeier: Grundlagen der Informatik. Nürnberger Hochschulsripten für Technik und Wirtschaft. Nürnberg, 2001, Nano-Verlag.

Leistungsnachweis

Schriftliche Prüfung (90 min)

Mathematik I (Algebra)

Studiengang	Bachelor Informatik 1. Studienabschnitt Theoretische Grundlagen
Modul	Mathematik I (Algebra)
Modulverantwortliche	Harald Stieber, Herbert Urban, Alexander Hufnagel, Elke Wilczok, Christian Scherr
Dozent(en)	Matthias Börger, Alexander Hufnagel, Christian Scherr, Herbert Urban
Vorkenntnisse	Schulwissen
Arbeitsaufwand	210 Stunden, davon: 95 Stunden Präsenzzeit 95 Stunden Vor- und Nachbereitung des Lehrstoffs 20 Stunden für Prüfungsvorbereitung
Leistungspunkte	7
Semesterwochenstunden	6
Veranstaltungstyp	4 SWS Vorlesung, 2 SWS Übung
Semesterturnus	Wintersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Vermittlung mathematischer Grundlagen
Lernziel	Begriffe und Strukturen aus verschiedenen grundlegenden Gebieten der Mathematik (Zahlen, Aussagenlogik, Mengen, Relationen und Funktionen) definieren und verstehen. Methoden der Kombinatorik zur Lösung von Zählproblemen einsetzen. Begriffe und Aussagen der elementaren Zahlentheorie aufstellen und Verständnis für spätere Anwendungen (insbesondere in der Kryptographie) erlangen. Rechenverfahren im Umgang mit komplexen Zahlen einüben und durchführen. Grundlegende Begriffe der Algebra kennen und in ihrer Bedeutung verstehen.
Schlüsselqualifikation	Fähigkeit zu abstraktem mathematischen Denken und Schließen, Fähigkeit zu komplexem Denken
Lehrinhalte	Zahlen, Aussagenlogik und Mengen, Relationen und Funktionen, Kombinatorik, Elemente der Zahlentheorie, komplexe Zahlen, Algebra (Gruppen, Ringe, Körper)
Literatur	G. Teschl, S. Teschl: Mathematik für Informatiker (Springer)
Leistungsnachweis	Schriftliche Prüfung (90 min.)

Mathematik II (Analysis)

Studiengang	Bachelor Informatik 1. Studienabschnitt Theoretische Grundlagen
Modul	Mathematik II (Analysis)
Modulverantwortliche	Harald Stieber, Herbert Urban, Alexander Hufnagel, Elke Wilczok, Christian Scherr
Dozent(en)	Jürgen Bolik, Matthias Börger, Alexander Hufnagel, Christian Scherr, Herbert Urban
Vorkenntnisse	Schulwissen, Mathematik I
Arbeitsaufwand	210 Stunden, davon: 95 Stunden Präsenzzeit 95 Stunden Vor- und Nachbereitung des Lehrstoffs 20 Stunden für Prüfungsvorbereitung
Leistungspunkte	7
Semesterwochenstunden	6
Veranstaltungstyp	4 SWS Vorlesung, 2 SWS Übung
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Vermittlung mathematischer Grundlagen
Lernziel	Grundlegende Begriffe und Strukturen der Linearen Algebra und der Analysis definieren und verstehen. Methoden des Rechnens mit Vektoren, Matrizen, Folgen und Reihen einüben und durchführen. Rechenverfahren der Differential- und Integralrechnung benutzen und in ausgewählten Problemstellungen anwenden. Differenzen- und Differentialgleichungen verstehen und Lösungsmethoden durchführen.
Schlüsselqualifikation	Fähigkeit zu abstraktem mathematischen Denken und Schließen, Fähigkeit zu komplexem Denken
Lehrinhalte	Lineare Algebra, Folgen und Reihen, Differential- und Integralrechnung, Differenzengleichungen, Differentialgleichungen
Literatur	<ul style="list-style-type: none">• G. Teschl, S. Teschl: Mathematik für Informatiker (Springer)• K. Meyberg, P. Vachenauer: Höhere Mathematik (Springer)
Leistungsnachweis	Schriftliche Prüfung (90 min.)

Theoretische Informatik

Studiengang	Bachelor Informatik 1. Studienabschnitt Theoretische Grundlagen
Modul	Theoretische Informatik
Modulverantwortliche	Thomas Fuhr, Alexander Kröner
Dozent(en)	Alexander Kröner, Hans Löhr
Vorkenntnisse	Schulwissen, Programmieren I
Arbeitsaufwand	210 Stunden, davon: 95 Stunden Präsenzzeit 115 Stunden Vor- und Nachbereitung des Lehrstoffs, Lösung der Übungsaufgaben und Prüfungsvorbereitung
Leistungspunkte	7
Semesterwochenstunden	6
Veranstaltungstyp	4 SWS Vorlesung mit 2 SWS Übung
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Schaffen von Grundlagen
Lernziel	Verständnis grundlegender Konzepte formaler Sprachen, der Automatentheorie sowie formaler Grammatiken und ihrer Zusammenhänge. Einordnung formaler Sprachen in die Chomsky-Hierarchie. Fähigkeit zur Abgrenzung regulärer und nicht-regulärer Sprachen. Analysieren von Automaten und Chomsky-Grammatiken. Formulieren von Grammatiken, regulären Ausdrücken und Automaten für formale Sprachen. Anwenden von Transformationen zur Überführung dieser Repräsentationen formaler Sprachen in äquivalente Modelle. Kenntnis des Berechenbarkeitsbegriffs und ausgewählter Entscheidbarkeitsprobleme.
Schlüsselqualifikation	Fähigkeit zu abstraktem mathematischen Denken und Schließen, Fähigkeit zur Bildung und zum Verständnis präziser formaler Modelle
Lehrinhalte	Endliche Automaten und formale Sprachen, Chomsky-Grammatiken und -Hierarchie, Kellerautomaten, Turingautomaten, Determinismus vs. Nichtdeterminismus, Berechenbarkeit und Entscheidbarkeit
Literatur	<ul style="list-style-type: none">• Dirk. W. Hoffmann: Theoretische Informatik, 3. Auflage, 2015, Carl Hanser Verlag.• G. Vossen, K.-U. Witt: Grundkurs theoretische Informatik: Eine anwendungsbezogene Einführung, 6. Auflage. Braunschweig, 2016, Springer Vieweg.• U. Hedstück: Einführung in die theoretische Informatik: Formale Sprachen und Automatentheorie, 5. Auflage. 2012, Oldenbourg Wissenschaftsverlag.• J.E.Hopcroft & R. Motwani & J.D.Ullman: Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie; 3. Auflage, 2011, Pearson Studium.• H. Knebl: Informationsverarbeitende Systeme. Nürnberger Hochschulschriften für Technik und Wirtschaft. Nürnberg, 2003, Nano-Verlag.

Programmieren I

Studiengang	Bachelor Informatik 1. Studienabschnitt Praktische Informatik
Modul	Programmieren I
Modulverantwortliche	Uwe Wienkop, Friedhelm Stappert, Jens Albrecht
Dozent(en)	Jens Albrecht, Friedhelm Stappert, Uwe Wienkop
Vorkenntnisse	Schulwissen
Arbeitsaufwand	210 Stunden, davon: 95 Stunden Präsenzzeit, 115 Stunden Vor- und Nachbereitung und Einübung des Lehrstoffs und Prüfungsvorbereitung
Leistungspunkte	7
Semesterwochenstunden	6
Veranstaltungstyp	4 SWS Vorlesung mit 2 SWS Übung
Semesterturnus	Wintersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Nahezu alle Berufsfelder für Absolventen und damit sehr viele Module der Studienprogramme der Studiengänge Informatik, Medieninformatik und Wirtschaftsinformatik haben mit der Anwendungsentwicklung von Software zu tun. Das Modul Programmieren I schafft dafür Grundlagen.
Lernziel	Nach dem erfolgreich Abschluss des Moduls Programmieren I sind die Studierenden in der Lage <ul style="list-style-type: none">• grundlegende Kontroll- und Datenstrukturen einer Programmiersprache sicher einzusetzen• einfache algorithmische Probleme zu analysieren und Lösungen dafür zu implementieren• wesentliche Konzepte objektorientierter Programmierung zu erläutern und anzuwenden• Elemente von Programmiersprachen wie Syntax, Namensbindung, Typsystem, Speicherstrukturen, Funktionsaufrufe und Parameterübergabe zu erkennen und zu erklären
Schlüsselqualifikation	<ul style="list-style-type: none">• Fähigkeit zu komplexem Denken• Ausdauer bei Problemlösungen
Lehrinhalte	<ul style="list-style-type: none">• Begriffe zur Datenverarbeitung, Syntax und Semantik der Sprachelemente: Ablaufstrukturen, Datenstrukturen, Objekte, Module, Iteration und Rekursion, Zeiger.• Entwicklungsmethoden: Entwicklungsumgebung, Entwicklung und Darstellung von Daten- und Ablaufstrukturen, strukturierter Entwurf und Implementierung, Dokumentation, Test.
Literatur	<ul style="list-style-type: none">• Kühnel, Andreas, <i>Visual C# 2012</i>, Rheinwerk Computing• Doberenz, Walter, <i>Visual C# 2015</i>, Hanser Verlag

- Hanspeter Mössenböck: *Kompaktkurs C# 6.0*, dpunkt Verlag
- Skript.

Leistungsnachweis	Schriftliche Prüfung (90 min)
Hilfsmittel	6 selbst erstellte Seiten
Zulassungsvoraussetzung	Erfolgreiche Teilnahme am Praktikum.

Programmieren II

Studiengang	Bachelor Informatik 1. Studienabschnitt Praktische Informatik
Modul	Programmieren II
Modulverantwortliche	Uwe Wienkop, Timo Götzelmann, Bartosz von Rymon Lipinski
Dozent(en)	Tobias Bocklet, Timo Götzelmann, Bartosz von Rymon Lipinski, Uwe Wienkop
Vorkenntnisse	Programmieren I
Arbeitsaufwand	210 Stunden, davon: 95 Stunden Präsenzzeit, 115 Stunden Vor- und Nachbereitung und Einübung des Lehrstoffs und Prüfungsvorbereitung
Leistungspunkte	7
Semesterwochenstunden	6
Veranstaltungstyp	4 SWS Vorlesung mit 2 SWS Übung
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Nahezu alle Berufsfelder für Absolventen und damit sehr viele Module der Studiengänge Informatik, Medieninformatik und Wirtschaftsinformatik haben mit der Anwendungsentwicklung von Software zu tun. Das Modul Programmieren II führt die in Programmieren I gelegten Grundlagen um i.w. objektorientierte Aspekte fort.
Lernziel	Vertiefung der Fähigkeiten, die in Programmieren I erworben wurden.
Schlüsselqualifikation	Fähigkeit zu komplexem Denken Ausdauer bei Problemlösungen
Lehrinhalte	Fortsetzung der Lehrinhalte von Programmieren I: dynamische Datenstrukturen, insbesondere verkettete Listen, Operatoren, Nutzung von Klassenbibliotheken, Ausnahmen und ihre Behandlung.
Literatur	<ul style="list-style-type: none">• Kühnel, Andreas, <i>Visual C# 2012</i>, Rheinwerk Computing• Doberenz, Walter, <i>Visual C# 2015</i>, Hanser Verlag• Hanspeter Mössenböck: <i>Kompaktkurs C# 6.0</i>, dpunkt Verlag• Skript.
Leistungsnachweis	Schriftliche Prüfung (90 min)

Software Engineering

Studiengang	Bachelor Informatik 1. Studienabschnitt Praktische Informatik
Modul	Software Engineering
Modulverantwortliche	Alfred Holl, Ramin Tavakoli Kolagari, Bartosz von Rymon Lipinski, Korbinian Riedhammer, Ronald Petrlc
Dozent(en)	Ronald Petrlc
Vorkenntnisse	Programmieren
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenzzeit, 85 Stunden Vor- und Nachbereitung des Lehrstoffs, Bearbeitung der Übungs- und Praktikumsaufgaben und Prüfungsvorbereitung
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	2 SWS Vorlesung, 2 SWS Übung und Praktikum
Semesterturnus	Winter- und Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	In den Berufsfeldern Anwendungsentwicklung und Entwicklung systemnaher Software gehört das Software Engineering zum unverzichtbaren Handwerkszeug.
Lernziel	Bei Abschluss des Lernprozesses wird der erfolgreiche Studierende in der Lage sein, Ziele, Methoden, Techniken und Verfahren des Software Engineering zu kennen und diese für konkrete Beispiele anwenden zu können. Die Studierenden werden Prozesse des Software Engineering verstehen und ausgewählte Techniken und Methoden für verschiedene Entwicklungsphasen auswählen können. Im Rahmen des begleitenden Praktikums werden sie teambasiert eine Beispielanwendung bzw. betriebliches Informationssystem im Rahmen eines vorgegeben Projektgerüsts prototypisch entwickeln können.
Schlüsselqualifikation	Fähigkeit zu komplexem Denken, Analyse und Klassifikation von Problemen.
Lehrinhalte	<ul style="list-style-type: none">• Probleme der industriellen Softwareerstellung;• Phasenmodelle;• Methoden zur Anforderungsspezifikation; Entwurfsmethoden;• Methoden zur Systemkonstruktion; Systemintegration und Test; Software-Ergonomie;• Qualitätssicherung; Softwaremetriken; Projektmanagement; DV-gestützte Entwicklungsumgebungen;• Aufwandsschätzungen; Wirtschaftlichkeitsbetrachtungen.
Literatur	<ul style="list-style-type: none">• C. Rupp, S. Queins, UML2 glasklar, 2012, Hanser.• G. Pomberger, W. Pree, Software Engineering, München, 2004, Hanser.• I. Sommerville, Software Engineering, München, 2012, Pearson Studium.
Leistungsnachweis	Schriftliche Prüfung (90 min). Erfolgreiche Teilnahme am Praktikum ist Voraussetzung für die Zulassung zur Prüfung.

Hilfsmittel

Zulassungsvoraussetzung

Grundlagen der Betriebswirtschaftslehre

Studiengang	Bachelor Informatik 1. Studienabschnitt Grundlagen der Betriebswirtschaftslehre
Modul	Grundlagen der Betriebswirtschaftslehre
Modulverantwortliche	Jürgen Badura, Patricia Brockmann, Wolfgang Bremer
Dozent(en)	Dina Barbian, Patricia Brockmann, Heidi Schuhbauer
Vorkenntnisse	Schulwissen
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenz, 85 Stunden Vor- und Nachbereiten des Lehrstoffs
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	4 SWS Vorlesung
Semesterturnus	Wintersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Betriebswirtschaftliche Grundlagen
Lernziel	Grundkenntnisse über Aufbau und Funktionen von Betrieben
Schlüsselqualifikation	Denken in wirtschaftlichen Zusammenhängen
Lehrinhalte	Wirtschaftswissenschaftliche Grundbegriffe; Aufbau und Funktionen von Betrieben (Standortwahl, Rechtsformen, Produktionsfaktoren); Grundbegriffe des betrieblichen Rechnungswesens.
Literatur	Straub, T.: Einführung in die Allgemeine Betriebswirtschaftslehre, 2. Auflage, Pearson 2014.
Leistungsnachweis	Schriftliche Prüfung (90 Min.)

Englisch I/II

Studiengang	Bachelor Informatik 1. Studienabschnitt Englisch
Modul	Englisch I/II
Modulverantwortliche	Thomas Huke
Dozent(en)	William Coleman, Joseph Raimond
Vorkenntnisse	Schulwissen
Arbeitsaufwand	120 Stunden, davon: 65 Stunden Präsenz 55 Stunden Vor- und Nachbereitung des Lehrstoffs und Prüfungsvorbereitung
Leistungspunkte	4
Semesterwochenstunden	4
Veranstaltungstyp	4 SWS seminaristischer Unterricht
Semesterturnus	Winter- bzw. Sommersemester
Unterrichtssprache	Kurs nur in Englisch
Beitrag zu den Zielen des Studiums	Die Studierenden identifizieren und erklären technische Originalliteratur und wenden die technische Sprache kontextgerecht in Wort und Schrift an.
Lernziel	Die Studierenden erwerben die Fähigkeit, gesprochenes und geschriebenes Englisch mit allgemeinsprachlichen und fachlichen Inhalten zu verstehen, sowie sich in der Fremdsprache mündlich und schriftlich korrekt auszudrücken.
Schlüsselqualifikation	Fremdsprachenkompetenz
Lehrinhalte	Fachbezogene Texte; allgemeine sowie fachbezogene Korrespondenz und Konversation; Erweiterung und Festigung der Vokabelkenntnisse; Umgang mit Hilfsmitteln
Literatur	<ul style="list-style-type: none">• Bruckenmaier, Stephan. 2008. <i>Grammatik kurz und bündig – Englisch</i>. Stuttgart: Klett Verlag (Pons).• Clark, David. 1997. <i>Englisch Grammatik</i>. Pocket Teacher. Berlin: Cornelsen Verlag.• Hoffmann, Hans & Marion Hoffmann. 2006. <i>Kurzgrammatik Englisch. Zum Nachschlagen und Üben</i>. München: Max Hueber.• Jacob, Rainer. 2008. <i>Kurzgrammatik. Kompakt-Wissen Englisch</i>. Freising: Stark Verlag.• Walther, Lutz. 2008. <i>Kurzgrammatik Englisch</i>. Berlin: Langenscheidt.• Weis, Erich. 1977. <i>Grund- und Aufbauwortschatz Englisch</i>. Stuttgart: Klett. (Nachdruck – 1986)
Leistungsnachweis	Schriftliche Prüfung (90 min)

Englisch III/IV (IN)

Studiengang	Bachelor Informatik 1. Studienabschnitt Englisch
Modul	Englisch III/IV (IN)
Modulverantwortliche	Thomas Huke
Dozent(en)	William Coleman
Vorkenntnisse	Schulwissen, Inhalt von Englisch I/II
Arbeitsaufwand	120 Stunden, davon: 65 Stunden Präsenz 55 Stunden Vor- und Nachbereitung des Lehrstoffs
Leistungspunkte	4
Semesterwochenstunden	4
Veranstaltungstyp	4 SWS seminaristischer Unterricht
Semesterturnus	Winter- bzw. Sommersemester
Unterrichtssprache	Kurs nur in Englisch
Beitrag zu den Zielen des Studiums	Die Studierenden setzen sich analytisch und hermeneutisch mit technischen und alltagspraktischen Texten mündlich und schriftlich auseinander.
Lernziel	Die Studierenden erwerben die Fähigkeit, gesprochenes und geschriebenes Englisch mit alltagspraktischen und fachlichen Inhalten anhand komplexer technischer und alltagspraktischer Texte zu analysieren und diskutieren. Besonderer Wert wird auf das Hörverstehen und den der Situation angemessenen schriftlichen Ausdruck gelegt.
Schlüsselqualifikation	Fremdsprachenkompetenz
Lehrinhalte	- Fachbezogene Texte - Allgemeine sowie fachbezogene Korrespondenz und Konversation - Erweiterung und Festigung der Vokabelkenntnisse - Umgang mit Hilfsmitteln.
Literatur	<ul style="list-style-type: none">• Bruckenmaier, Stephan. 2008. <i>Grammatik kurz und bündig – Englisch</i>. Stuttgart: Klett Verlag (Pons).• Clark, David. 1997. <i>Englisch Grammatik</i>. Pocket Teacher. Berlin: Cornelsen Verlag.• Hoffmann, Hans & Marion Hoffmann. 2006. <i>Kurzgrammatik Englisch. Zum Nachschlagen und Üben</i>. München: Max Hueber.• Jacob, Rainer. 2008. <i>Kurzgrammatik. Kompakt-Wissen Englisch</i>. Freising: Stark Verlag.• Walther, Lutz. 2008. <i>Kurzgrammatik Englisch</i>. Berlin: Langenscheidt.• Weis, Erich. 1977. <i>Grund- und Aufbauwortschatz Englisch</i>. Stuttgart: Klett. (Nachdruck – 1986)
Leistungsnachweis	Schriftliche Prüfung (90 min)

Allgemeinwissenschaftliche Wahlpflichtmodule

Studiengang	Bachelor Informatik 2. Studienabschnitt allgemeines Modul
Modul	Allgemeinwissenschaftliche Wahlpflichtmodule
Vorkenntnisse	Die Studierenden der Informatik und Wirtschaftsinformatik können die von ihnen zu erbringenden Allgemeinwissenschaftlichen Wahlpflichtfächer aus den Vorlesungen des Studium Generale der Georg-Simon-Ohm-Hochschule Nürnberg insbesondere aus den folgenden Teilbereichen auswählen: Recht, Wirtschaft, moderne Sprachen, Technik, Gesellschaft, Geschichte, Politik, Persönlichkeitsbildung, Psychologie, Philosophie und Literatur. In der Regel sind keine Vorkenntnisse erforderlich, außer z. B. in Sprachkursen für Fortgeschrittene.
Arbeitsaufwand	60 Stunden, davon: 30 Stunden Präsenz und 30 Stunden Vor- und Nachbereitung des Stoffs und Prüfungsvorbereitung
Leistungspunkte	2
Semesterwochenstunden	2
Veranstaltungstyp	Je nach Veranstaltung Vorlesung, Seminar, etc.
Semesterturnus	Winter- und Sommersemester
Unterrichtssprache	Kurs in Deutsch und/oder Englisch (bitte nachfragen)
Beitrag zu den Zielen des Studiums	Ziel des Studiums ist ein berufsfähiger Abschluss. Berufsfähigkeit bedeutet immer auch, den Blick über den Tellerrand heben zu können. Im Rahmen der Sprachkurse ist darüberhinaus oft ein unmittelbarer direkter Vorteil für die Berufsfähigkeit zu erkennen.
Lernziel	Die Erkenntnis, dass das eigene Fachgebiet nicht isoliert existieren kann, sondern dem Lebensumfeld zugehört.
Schlüsselqualifikation	Fachspezifisch; häufig Sozialkompetenz (Kooperationsfähigkeit, Teamfähigkeit), Fremdsprachenkompetenz, Präsentationskompetenz
Lehrinhalte	Als allgemeinwissenschaftliche Wahlpflichtfächer können alle an der Hochschule angebotenen Lehrveranstaltungen gewählt werden, soweit sie nicht Pflichtfächer oder fachwissenschaftliche Wahlpflichtfächer des Studiengangs Informatik sind beziehungsweise in der Ausschlussliste des Studiengangs geführt werden.
Literatur	Fachspezifisch
Leistungsnachweis	Nach der Zuteilung einer Studienarbeit oder der Ablegung einer Prüfungsleistung kann ein allgemeinwissenschaftliches Wahlpflichtfach unabhängig vom Bestehen nicht mehr gewechselt werden. Der Leistungsnachweis ist fachspezifisch und wird im Katalog der Fachwissenschaftlichen Wahlpflichtfächer vor Modulbeginn angegeben.

Algorithmen und Datenstrukturen

Studiengang	Bachelor Informatik 2. Studienabschnitt Theoretische Grundlagen
Modul	Algorithmen und Datenstrukturen
Modulverantwortliche	Thomas Fuhr, Peter Trommler
Dozent(en)	Peter Trommler
Vorkenntnisse	Inhalte der Lehrveranstaltungen des ersten Studienjahres, insbesondere Mathematik, Grundlagen der Informatik, Theoretische Informatik und Programmieren
Arbeitsaufwand	210 Stunden, davon: 95 Stunden Präsenzzeit, 125 Stunden Vor- und Nachbereitung des Lehrstoffs, Lösung der Praktikumsaufgaben und Prüfungsvorbereitung
Leistungspunkte	7
Semesterwochenstunden	6
Veranstaltungstyp	4 SWS Vorlesung, 2 SWS Übung/Praktikum
Semesterturnus	Winter- und Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Veranstaltung mit Grundlagen- und Querschnittcharakter. Behandelte Themen sind grundlegend für das Verständnis der Realisierung von Standardsoftware (z.B. Datenbanken, Systemsoftware) sowie für die Entwicklung adäquater Softwarelösungen aller Art.
Lernziel	Grundlegende Datenstrukturen unter Einbeziehung externer Speichermedien kennen; zugehörige Algorithmen und ihre Zusammenhänge verstehen sowie diese korrekt für konkrete Beispiele ausführen können; die Aspekte, Korrektheit, Komplexität und Effizienz von Algorithmen sowie übliche Entwurfsprinzipien kennen; einfache Algorithmen analysieren und ihren Aufwand mathematisch beschreiben und kategorisieren bzw. deren Korrektheit beweisen können; Algorithmen hinsichtlich ihres Aufwands einordnen und vergleichend bewerten können.
Schlüsselqualifikation	Fähigkeit zu komplexem, abstraktem mathematischen Denken und Schließen, selbständiges Arbeiten, Analyse und Klassifikation von Problemen, kreatives Problemlösen, Ausdauer bei Problemlösungen
Lehrinhalte	Die Aspekte Korrektheit, Komplexität und Effizienz von Algorithmen sowie die Entwurfsprinzipien für Algorithmen. Inhalt sind unter anderem die Themen Sortieren und Suchen, Suchbäume, Hashverfahren, Organisation von Wörterbüchern und Graphalgorithmen. Umsetzung ausgewählter Algorithmen auf Basis einer Programmiersprache.
Literatur	<ul style="list-style-type: none">• T.H.Cormen, C.E.Leiserson, R.L.Rivest, C.Stein: Algorithmen – Eine Einführung, 4. Aufl., 2013, De Gruyter Oldenbourg• T. Ottmann, P. Widmayer: Algorithmen und Datenstrukturen, 5. Aufl. 2012, Spektrum Akademischer Verlag.• V. Heun: Grundlegende Algorithmen: Einführung in den Entwurf und die Analyse effizienter Algorithmen, 2.Aufl., Braunschweig, Wiesbaden, 2003, Vieweg.• D.E. Knuth: The Art of Computer Programming, Volume 1+3: Fundamental

Algorithms + Searching and Sorting. Reading, MA, 1998, Addison-Wesley Publishing Company.

- R. Motwani, P. Raghavan: Randomized Algorithms. Cambridge, 1995, Cambridge University Press.
- H. Knebl: Algorithmen und Datenstrukturen. Nürnberg, 2005, Skript.

Leistungsnachweis

Schriftliche Prüfung (90 min). Zulassungsvoraussetzung für die Teilnahme an der schriftlichen Prüfung ist die erfolgreiche Teilnahme am Praktikum.

Kryptographie und Informationssicherheit

Studiengang	Bachelor Informatik 2. Studienabschnitt Theoretische Grundlagen
Modul	Kryptographie und Informationssicherheit
Modulverantwortliche	Hans Löhr
Dozent(en)	Ronald Petrlc
Vorkenntnisse	Lehrveranstaltungen des ersten Studienjahres, insbesondere Mathematik I,II, Grundlagen der Informatik, Programmieren.
Arbeitsaufwand	210 Stunden, davon: 90 Stunden Präsenzzeit, 120 Stunden für Vor- und Nachbereitung des Lehrstoffs, Bearbeitung der Übungsaufgaben und Prüfungsvorbereitung.
Leistungspunkte	7
Semesterwochenstunden	6
Veranstaltungstyp	4 SWS Vorlesung 2 SWS Übung.
Semesterturnus	Wintersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Mit der zunehmenden Vernetzung von Systemen und dem rapiden Wachstum elektronischer Kommunikation erlangen Fragen der Informationssicherheit eine immer größere praktische Bedeutung. Informationen, die über das Internet ausgetauscht werden, sollen vertraulich bleiben und gegen Manipulation geschützt werden. E-Business oder E-Government-Anwendungen benötigen digitale Unterschriften. Die Computersysteme eines Unternehmens müssen vor unberechtigten Zugriffen geschützt werden. Deshalb ist es für jeden Informatiker wichtig, die Probleme der Informationssicherheit zu kennen, ebenso wie die grundlegenden Techniken, die von der Kryptographie zur Lösung der Probleme bereitgestellt werden.
Lernziel	Die Bedeutung der Informationssicherheit kennen und die grundlegenden Schutzziele der IT-Sicherheit verstehen und abgrenzen können; grundlegende Sicherheitstechniken aus der symmetrischen und asymmetrischen Kryptographie verstehen und anwenden, zwischen ihnen unterscheiden und sie zum Erreichen vorgegebener Schutzziele auswählen können; Angriffe und Sicherheitsanalysen verstehen, Sicherheitseigenschaften bewerten können.
Schlüsselqualifikation	Fähigkeit zu komplexem, abstraktem Denken und Schließen, Analyse und Klassifikation von Problemen, kreatives Problemlösen.
Lehrinhalte	<ul style="list-style-type: none">• Ziele in der Informationssicherheit: Vertraulichkeit, Datenintegrität, Authentifizierung des Ursprungs von Daten, Authentifizierung von Entitäten, Verbindlichkeit;• Symmetrische Verschlüsselungsverfahren: Stromchiffre, Blockchiffre (z. B. DES, AES), Operationsmodi;• Kryptographische Hashfunktionen, Message Authentication Codes (MAC);• Asymmetrische Verfahren / Public Key - Kryptographie: Einwegfunktionen, Verschlüsselung, digitale Signatur, RSA-, ElGamal/DSA-Verfahren; Angriffe; optimal asymmetric encryption padding OAEP; Kryptographische Protokolle für den Schlüsselaustausch;

- Transport Layer Security (SSL/TLS).

Literatur

- H. Delfs, H. Knebl, Introduction to Cryptography – Principles and Applications. Springer Book Series on Information Security and Cryptography. Berlin, Heidelberg, New York: Springer-Verlag, 3rd edition, 2015.
- J.A. Buchmann: Introduction to Cryptography. 2nd edition. Berlin, Heidelberg, New York: Springer-Verlag, 2004.
- D.R. Stinson: Cryptography – Theory and Practice. Boca Raton, New York, London, Tokyo: CRC-Press, 2002.
- J. Schwenk: Sicherheit und Kryptographie im Internet. 4. Auflage, Springer Vieweg, 2014.

Leistungsnachweis

Schriftliche Prüfung (90 min).

Mathematik III (Statistik)

Studiengang	Bachelor Informatik 2. Studienabschnitt Theoretische Grundlagen
Modul	Mathematik III (Statistik)
Modulverantwortliche	Hans Delfs, Harald Stieber, Alexander Hufnagel
Dozent(en)	Jürgen Bolik, Alexander Hufnagel, Florian Loos, Christian Scherr, Edgar Wermuth
Vorkenntnisse	Mathematik I und II
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenzzeit, 65 Stunden Vor- und Nachbereitung des Lehrstoffs und Prüfungsvorbereitung 20 Stunden für Prüfung und Prüfungsvorbereitung
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	4 SWS Vorlesung
Semesterturnus	Winter- und Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Wahrscheinlichkeitsrechnung und Statistik sind wichtige Grundlagen für verschiedene Gebiete der Informatik.
Lernziel	Grundlegende Methoden der Wahrscheinlichkeitsrechnung und der beschreibenden und schließenden Statistik verstehen und eigenständig anwenden, die Voraussetzungen ihrer Anwendung prüfen, statistische Modelle vergleichen und zur Anwendungssituation passende auswählen, Ergebnisse sinnvoll interpretieren können.
Schlüsselqualifikation	Fähigkeit zu abstraktem mathematischen Denken und Schließen
Lehrinhalte	Deskriptive Statistik: Lage- und Streuungsmaße, Korrelation und Regression; Elemente der Wahrscheinlichkeitsrechnung: Zufallsvariable, Bedingte Wahrscheinlichkeiten, Wahrscheinlichkeitsverteilungen, Poisson-Prozesse; Induktive Statistik: Punktschätzung, Intervallschätzung, Konfidenzintervalle, Testen von Hypothesen
Literatur	G. Teschl, S. Teschl: Mathematik für Informatiker, Band 2. 3. Auflage. Berlin, Heidelberg: Springer 2014. J. Lehn, H. Wegmann: Einführung in die Statistik. 5. Auflage. Wiesbaden: Teubner 2006.
Leistungsnachweis	Schriftliche Prüfung (90 min)

Datenbanken

Studiengang	Bachelor Informatik 2. Studienabschnitt Softwaremethoden und -technologien
Modul	Datenbanken
Modulverantwortliche	Patricia Brockmann, Alfred Holl, Jens Albrecht
Dozent(en)	Jens Albrecht, Patricia Brockmann, Alfred Holl
Vorkenntnisse	Lehrveranstaltungen des ersten Studienjahres, insbesondere Grundlagen der Informatik, Programmieren.
Arbeitsaufwand	180 Stunden, davon: 90 Stunden Präsenzzeit, 90 Stunden für Vor- und Nachbereitung des Lehrstoffs, Bearbeitung der Übungs- und Praktikumsaufgaben und Prüfungsvorbereitung
Leistungspunkte	6
Semesterwochenstunden	6
Veranstaltungstyp	4 SWS Vorlesung, 2 SWS Übung mit Praktikum
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	In der Anwendungsentwicklung und im Systemmanagement werden gute Kenntnisse über die Funktionsweise und Leistungsfähigkeit von Datenbanksystemen benötigt. Methoden des systematischen Entwurfs von Datenmodellen für Datenbanken zu beherrschen, Datenbanken implementieren und den Zugriff auf die gespeicherten Daten aus Anwendungssystemen realisieren zu können, sind wesentliche Fähigkeiten eines Informatikers.
Lernziel	Nach dem erfolgreichen Abschluss des Moduls Datenbanken sind die Studierenden in der Lage <ul style="list-style-type: none">• die Bedeutung von Datenbanken in Software-Systemen zu bewerten• verschiedene Datenbankmodelle voneinander abzugrenzen• die Grundlagen relationaler Datenbanksysteme zu erklären• konzeptionelle Datenmodelle zu erstellen und in normalisierte relationale Datenbankschemata zu überführen• komplexe SQL-Anfragen und SQL-Änderungsoperationen zu formulieren• die Qualität eines entworfenen Datenbankschemas zu bewerten• die Eignung eines relationalen Datenbanksystems für ein gegebenes Problem einzuschätzen
Schlüsselqualifikation	Fähigkeit zu komplexem, abstraktem mathematischen Denken und Schließen, Analyse und Klassifikation von Problemen, kreatives Problemlösen.
Lehrinhalte	Aufbau und Funktionen eines Datenbanksystems; objektorientiertes und relationales Datenmodell, objektrelationale Abbildung; relationale, objektrelationale, föderative, verteilte Datenbanksysteme; Datenmodellierung und Datenbankentwurf, Referenzmodelle, Unternehmensdatenmodelle, Datenintegration; Datenbankabfragesprachen: Datendefinition und –manipulation mit der Structured Query Language SQL; Schnittstellen für die Anwendungsprogrammierung; Transaktionsmanagement, Concurrency und Recovery. Einführung in NoSQL-

Datenbanken und Big Data.

Literatur

- Kemper, A.; Eickler, A.: Datenbanksysteme - Eine Einführung, De Gruyter Studium, 2015
- Sattler, K.; Heuer, A.: Datenbanken – Konzepte und Sprachen. mitp, 2013
- Saake, G.; Heuer, K.; A. Sattler: Datenbanken – Implementierungstechniken. mitp, 2011
- Elmasri, R. , Navathe, S., "Grundlagen von Datenbanksystemen", Bachelorausgabe, Pearson 2010

Leistungsnachweis

Schriftliche Prüfung (90 min).

Hilfsmittel

10 selbst erstellte Seiten

Zulassungsvoraussetzung

Erfolgreiche Teilnahme am Praktikum.

Interaktive Systeme

Studiengang	Bachelor Informatik 2. Studienabschnitt Softwaremethoden und -technologien
Modul	Interaktive Systeme
Modulverantwortliche	Timo Götzelmann, Alexander Kröner
Dozent(en)	Timo Götzelmann, Alexander Kröner
Vorkenntnisse	Grundständige Programmierkenntnisse
Arbeitsaufwand	210 Stunden, davon: 90 Stunden Präsenz 120 Stunden Vor- und Nachbereitung des Lehrstoffs und Prüfungsvorbereitung
Leistungspunkte	7
Semesterwochenstunden	6
Veranstaltungstyp	4 SWS Vorlesung, 2 SWS Übung
Semesterturnus	Wintersemester
Unterrichtssprache	Kurs in Deutsch und Englisch
Beitrag zu den Zielen des Studiums	Die Studierenden lernen spezielle Problemstellungen der Entwicklung interaktiver Mensch-Computer Systeme kennen. Sie können wesentliche Anforderungen interaktiver Systeme identifizieren. Aufbauend auf grundständigen Programmierkenntnissen können sie damit selbstständig in Verbindung stehende Lösungsansätze entwickeln.
Lernziel	Kenntnis von Historie und Merkmalen interaktiver Systeme. Fähigkeit zur Klassifizierung solcher Systeme. Fähigkeit zur Beurteilung der Gebrauchstauglichkeit von Benutzerschnittstellen. Kenntnis von Entwicklungsprozess und Architekturmustern interaktiver Systeme. Fähigkeit zur Anwendung vorher genannter Kenntnisse zur Implementierung interaktiver Systeme.
Schlüsselqualifikation	Analyse und Klassifikation von Problemen, kreatives Problemlösen, fachübergreifendes Wissen, praktische Erfahrung
Lehrinhalte	<ul style="list-style-type: none">• Historische Entwicklung und Arten von Benutzerschnittstellen• Interaktionstechnologien und deren Interaktionstechniken• Graphische Dialogsysteme• Modelle und Architekturmuster für interaktive Systeme• Entwicklungsframeworks
Leistungsnachweis	Schriftliche Prüfung (90 min.)

IT-Anwendungen (Bachelor IN)

Studiengang	Bachelor Informatik 2. Studienabschnitt Softwaremethoden und -technologien
Modul	IT-Anwendungen (Bachelor IN)
Modulverantwortliche	Peter Trommler
Dozent(en)	Florian Gallwitz, Timo Götzelmann, Ronald Petrlc, Korbinian Riedhammer, Jörg Roth, Bartosz von Rymon Lipinski, Michael Zapf
Vorkenntnisse	Programmieren, Software Engineering, Datenbanken, Betriebssysteme, Rechnernetze, Rechnerkommunikation, Projektmanagement
Arbeitsaufwand	270 Stunden Projektarbeit, davon: 90 Stunden Präsenz (Projektbesprechungen, gemeinsam in der Projektgruppe zu leistende Diskussion der Ziele, Planungsarbeiten, Integration von Arbeitsergebnissen, Integrationstest, Präsentation des Projektergebnisses, Vortragsveranstaltungen).
Leistungspunkte	9
Semesterwochenstunden	6
Veranstaltungstyp	Projektarbeit
Semesterturnus	Winter- bzw. Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Das Projekt bereitet auf alle Berufsfelder vor, die mit der Entwicklung von Software verbunden sind.
Lernziel	Einblick in die Vielgestaltigkeit von Anwendungen der Informationstechnologie. Fähigkeit zur Umsetzung einer Problemstellung in eine IT-Lösung mit Teamarbeit.
Schlüsselqualifikation	Sozialkompetenz (Kooperationsfähigkeit, Teamfähigkeit), Organisationskompetenz, Projekterfahrung, Projektmanagement, Zeitmanagement, Ausdauer bei Problemlösungen.
Lehrinhalte	Die Studierenden bearbeiten in Teams von ca. 4 Mitgliedern jeweils ein Thema aus der angewandten Informatik. Sie werden dabei von einem Dozenten betreut. Typischerweise ist für eine ausgewählte Problemstellung eine IT-Lösung in Form einer Software-Anwendung zu planen, zu entwerfen und zu implementieren, oder es sind implementierte Lösungen zu beurteilen und anzupassen. Die Projektarbeit erstreckt sich über 2 Semester. Im ersten der beiden Semester analysiert das Team die gestellte Aufgabe, entwickelt einen Projektplan und beginnt mit der Umsetzung des Projekts, beispielsweise dem Entwurf oder einem Prototypen zu einer Softwarelösung. Im zweiten Semester wird das Projekt fertiggestellt, mit einem Bericht dokumentiert und dem Betreuer und Mitstudierenden in einem begleitenden Seminar vorgestellt.
Literatur	Abhängig vom Projektthema
Leistungsnachweis	Studienarbeit (bis 5 Seiten), Referat (60 Min.)

Programmiersprachen

Studiengang	Bachelor Informatik 2. Studienabschnitt Softwaremethoden und -technologien
Modul	Programmiersprachen
Modulverantwortliche	Christian Schiedermeier, Matthias Teßmann
Dozent(en)	Christian Schiedermeier
Vorkenntnisse	Stoff der Lehrveranstaltungen Programmieren, Theoretische Informatik. Praktische Tätigkeit
Arbeitsaufwand	210 Stunden, davon: 90 Stunden Präsenzzeit, 120 Stunden Vor- und Nachbereitung des Lehrstoffs und Prüfungsvorbereitung
Leistungspunkte	7
Semesterwochenstunden	6
Veranstaltungstyp	4 SWS Vorlesung, 2 SWS Übung
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Diese Veranstaltung führt zu einem vertieften Verständnis systemnaher Software (z. B. Compiler, virtuelle Maschinen) und dem adäquaten Einsatz von Programmiersprachen in der Anwendungsentwicklung (z. B. Sprachparadigmen, formale Sprachen, Code-Sicherheit).
Lernziel	Kenntnis wesentlicher Konzepte von Programmiersprachen und von Programmierparadigmen. Kenntnis der Übersetzung von prozeduralen und objektorientierten Programmiersprachen sowie ihrer Ausführung in modernen Laufzeitsystemen. Fähigkeit, Programme in unbekanntem Programmiersprachen zu verstehen und Bezüge zu bisherigen Kenntnissen herzustellen. Fähigkeit, die Eignung unterschiedlicher Programmierparadigmen und Programmiersprachen für verschiedene Anwendungsaufgaben zu untersuchen und zu beurteilen.
Schlüsselqualifikation	Fähigkeit zu komplexem Denken, Fähigkeit zu abstraktem mathematischen Denken und Schließen
Lehrinhalte	<ul style="list-style-type: none">• Syntaxanalyse, semantische Analyse, Codegenerierung• Werte, Typen, Namen, Bindungen, Speicherabbildung, Kontrollfluss, Typsysteme, prozedurale, Daten- und generische Abstraktion.• Prozedurale, objektorientierte und funktionale Programmierung, Skript-Sprachen.• Virtuelle Maschinen, Typsysteme zur Laufzeit, Komponenten, Speicherverwaltung, Code-Sicherheit.
Literatur	<ul style="list-style-type: none">• Michael L. Scott: Programming Language Pragmatics, Morgan Kaufmann, 2016• Keith D. Cooper, Linda Torczon: Engineering a Compiler, 2nd edition, Morgan Kaufman, 2011• Dick Grune et. al.: Modern Compiler Design, Springer, 2016
Leistungsnachweis	Schriftliche Prüfung (90 min)

Softwarearchitektur

Studiengang	Bachelor Informatik 2. Studienabschnitt Softwaremethoden und -technologien
Modul	Softwarearchitektur
Modulverantwortliche	Ramin Tavakoli Kolagari, Korbinian Riedhammer
Dozent(en)	Korbinian Riedhammer
Vorkenntnisse	Praktisches Studiensemester
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenzzeit, 85 Stunden Vor- und Nachbereitung des Lehrstoffs, Übungsaufgaben und Prüfungsvorbereitung.
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Vorlesung mit Übungen
Semesterturnus	Winter- bzw. Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Erfahrungsgemäß gehören der Entwurf und die Architektur großer Softwaresysteme zu den schwierigsten Aufgaben der praktischen Informatik. Diese Lehrveranstaltung bereitet die Studierenden darauf vor, indem sie architektonische Alternativen und Kriterien für ihre Bewertung kennen lernen.
Lernziel	Bei Abschluss des Lernprozesses wird der erfolgreiche Studierende in der Lage sein, Konzepte guter Softwarearchitektur, insbesondere Clean Code, Programmieridiome, Entwurfsmuster, Architekturmuster und Komponenten zu kennen und diese für konkrete Szenarien anwenden, die unterschiedlichen Prinzipien der Softwarearchitektur für den Entwurf eines Systems auswählen und kleine Beispielanwendung unter Vermeidung bekannter Architekturprobleme entwickeln zu können.
Schlüsselqualifikation	Fähigkeit zum Verständnis großer komplexer Systeme; Analyse und Klassifikation von Problemen; kreatives Problemlösen.
Lehrinhalte	Dieser Kurs vermittelt grundlegende Techniken zum Entwurf, zur Beschreibung und zur Implementierung von großen, modularen Softwaresystemen. Zentraler Begriff der Vorlesung sind Patterns, welche als Grundprinzipien und Bausteine einer Softwarearchitektur verstanden werden können. Der Kurs führt Patterns auf verschiedenen Ebenen eines Softwaresystems ein: Auf Implementierungsebene (Programming Idioms), auf Ebene verschiedener Softwaremodule, Klassen und deren Zusammenspiel (Design Patterns), und schließlich als prototypische Struktur für ganze Anwendungen (Architectural Patterns). Muster werden anhand von Programmbeispielen veranschaulicht. Es werden mehrere Programmierübungen im Labor bearbeitet.
Literatur	<ul style="list-style-type: none">• Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: "Design Patterns, Elements of Reusable Object-Oriented Software", Addison Wesley, 1994.• Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal: "Pattern-Oriented Software Architecture", Wiley, 1996.

- Alan Shalloway, James R. Trott: "Design Patterns Explained: A New Perspective on Object-Oriented Design", Addison Wesley, 2005.
- Matthias Geirhos: "Entwurfsmuster: Das umfassende Handbuch", Rheinwerk Computing, 2015.
- Mahboub Gharbi, Arne Koschel, Andreas Rausch, Gernot Starke: "Basiswissen für Softwarearchitekten", dpunkt.verlag, 2013.

Leistungsnachweis

Schriftliche Prüfung (90 Minuten)

Web-Programmierung

Studiengang	Bachelor Informatik 2. Studienabschnitt Softwaremethoden und -technologien
Modul	Web-Programmierung
Modulverantwortliche	Matthias Teßmann
Dozent(en)	Matthias Teßmann
Vorkenntnisse	Programmieren I und II, Software Engineering, Grundlagen der Informatik
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenz, 85 Stunden Vor- und Nachbereiten des Lehrstoffs sowie Übungsaufgaben/Praktikum
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	2 SWS Vorlesung, 2 SWS Übungen und studienbegleitendes Praktikum
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Nahezu alle Berufsfelder für Absolventen und damit sehr viele Module der Studienprogramme der Studiengänge Informatik, Medieninformatik und Wirtschaftsinformatik haben mit der Anwendungsentwicklung von Software zu tun. Das Modul Web-Programmierung führt die in Programmieren I und II gelegten Grundlagen mit einem besonderem Fokus auf Webanwendungen fort.
Lernziel	Durch das Modul Web-Programmierung lernen die Studierenden die wichtigsten Schlüsseltechniken des World Wide Web kennen. Dies umfasst sowohl client- als auch serverseitige Technologien, wie unter anderem HTML5, CSS3, JavaScript, HTTP, Webserver (Apache2, Nginx), PHP, REST-Schnittstellen und asynchrone Anfragen. Dadurch erlangen Sie ein Verständnis für die Zusammenhänge dieser und werden dazu befähigt, deren Vor- und Nachteile gegeneinander abzuwägen. Die vermittelten theoretischen Kenntnisse werden im Rahmen des Pflichtpraktikums durch die selbstständige Konzeption und Umsetzung einer vollständigen Client-Server-Web-Anwendung (z.B. Online-Shop, soziales Netzwerk, ...) modulbegleitend praktisch gefestigt. Dabei kommt die Lernform des problembasierten Lernens (PBL) zum Einsatz. Das Modul schließt mit der Vermittlung grundlegender Kenntnisse über die wichtigsten Sicherheitsrisiken von Web-Anwendungen.
Schlüsselqualifikation	Analyse und Klassifikation von Problemen, kreatives Problemlösen, selbständiges Arbeiten, Fähigkeit zu komplexem Denken, Fähigkeit zum Verständnis großer komplexer Systeme, Lernen am Beispiel, Transferfähigkeit
Lehrinhalte	Die Inhalte der Lehrveranstaltung sind u.a.: <ul style="list-style-type: none">• Grundlagen<ul style="list-style-type: none">◦ HTML5, CSS3, JavaScript• Client-Server Modell / Kommunikationsprotokolle und Datenformate<ul style="list-style-type: none">◦ HTTP/HTTPS/HTTP2◦ JSON• Front-End, Back-End und "Full-Stack"-Anwendungsentwicklung

- PHP, Application-Server, node.js
- Cookies und Sessions
- Asynchrone Kommunikation
- Browser- und Server-Architekturen
- Single-Page-Anwendungen

- Der Browser als universelle Anwendungsplattform
- Web-Anwendungssicherheit (OWASP Top 10)

Literatur

- The World Wide Web Consortium (W3C), Standards and Recommendations, Online: <http://w3.org>
- Jürgen Wolf, "HTML5 und CSS3: Das umfassende Handbuch.", 2. Auflage, Rheinwerk Computing, 2016
- David Flanagan, "JavaScript: The Definitive Guide", 6th. Ed.. O'Reilly, 2011
- Douglas Crockford, "JavaScript: The Good Parts", O'Reilly, 2008
- Steve Prettyman, "Learn PHP 7", Apress, 2016
- The Open Web Application Security Project, Documents and Recommendations, Online: <http://www.owasp.org>

Leistungsnachweis

Schriftliche Prüfung (90 Minuten), erfolgreiche Teilnahme am Praktikum für die Zulassung erforderlich

Hilfsmittel

Zulassungsvoraussetzung

Betriebssysteme

Studiengang	Bachelor Informatik 2. Studienabschnitt Systeminfrastrukturen
Modul	Betriebssysteme
Modulverantwortliche	Friedhelm Stappert, Axel Hein
Dozent(en)	Michael Zapf
Vorkenntnisse	Rechnersysteme, Grundlagen der Informatik, Theoretische Informatik, Programmieren.
Arbeitsaufwand	210 Stunden, davon: 90 Stunden Präsenzzeit, 120 Stunden Vor- und Nachbereitung des Lehrstoffs und Prüfungsvorbereitung.
Leistungspunkte	7
Semesterwochenstunden	6
Veranstaltungstyp	4 SWS Vorlesung mit 2 SWS Übung.
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Betriebssysteme gehören zu den Grundlagen von jeder Art von Computersystemen. Alle Einsatzmöglichkeiten eines Informatikers erfordern die Kenntnis grundlegender Betriebssystemkonzepte.
Lernziel	<ul style="list-style-type: none">• Einsicht in die theoretischen Grundlagen von Betriebssystemen; exemplarisch verdeutlicht an Hand konkreter Beispiele.• Aufbau und Architektur von Betriebssystemen verstehen und analysieren.• Einfache, betriebssystemspezifische Programme erstellen können (Systemprogrammierung).• Parallele Prozesse und Synchronisationsmechanismen verstehen und anwenden.• Fähigkeit, aktuelle Betriebssysteme und ihre Dienste einzuordnen und zu bewerten.
Schlüsselqualifikation	Fähigkeit zu komplexem Denken, Ausdauer bei Problemlösungen.
Lehrinhalte	<ul style="list-style-type: none">• Aufgaben und Aufbau von Betriebssystemen• Verwaltungsfunktionen (insbesondere Prozesse, Speicher, Dateien)• Kommunikationsfunktionen
Literatur	<ul style="list-style-type: none">• Eduard Glatz: Betriebssysteme – Grundlagen, Konzepte, Systemprogrammierung, dpunkt.verlag, 3. Auflage, Heidelberg, 2015• Andrew S. Tanenbaum: Moderne Betriebssysteme, 2016, Pearson• William Stallings: Operating Systems, Prentice Hall, 2008• M. Mitchell, J. Oldham, A. Samuel: Advanced Linux Programming. New Riders, 2001• Peter Mandl: Grundkurs Betriebssysteme. Springer 2013• Vorlesungsunterlagen
Leistungsnachweis	Schriftliche Prüfung (90 min)

Rechnerkommunikation und -netze

Studiengang	Bachelor Informatik 2. Studienabschnitt Systeminfrastrukturen
Modul	Rechnerkommunikation und -netze
Modulverantwortliche	Jörg Roth
Dozent(en)	Jörg Roth
Vorkenntnisse	Grundlagen der Informatik, Theoretische Informatik
Arbeitsaufwand	210 Stunden, davon: 90 Stunden Präsenzzeit, 120 Stunden für Vor- und Nachbereitung des Lehrstoffs, Bearbeitung der Übungsaufgaben und Prüfungsvorbereitung
Leistungspunkte	7
Semesterwochenstunden	6
Veranstaltungstyp	Vorlesung
Semesterturnus	Wintersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Kenntnisse der Funktionsweise von Kommunikationsnetzen als Basis der Kommunikation von Rechnern und verteilter Information, Fachbezogene Englisch-Kenntnisse
Lernziel	Bei Abschluss des Lernprozesses wird der erfolgreiche Studierende in der Lage sein, grundlegende Konzepte und Mechanismen in Rechnernetzen, insbesondere im Umgang mit Netzwerk-Adressen verschiedener Ebenen und mit Mechanismen von Datagramm- und verbindungsorientierten Protokollen zu kennen und die grundlegenden Kommunikationsprobleme auf verschiedenen Schichten zu verstehen und Lösungen auszuwählen.
Schlüsselqualifikation	Fähigkeit zu komplexem Denken, Analyse und Klassifikation von Problemen. Fähigkeit zum Umgang mit englischsprachigen Standardtexten
Lehrinhalte	OSI-Architektur-Modell; Physische Übertragungsmedien; Kodierung, Sicherung von Rahmen, Medienzugriffsmethoden; Routing, Internet Protokoll (IP); Transportprotokolle (z.B. TCP und UDP); anwendungsnahen Schichten; Internetprotokolle; Realisierung von Diensten (mit z.B. Web Services); Infrastrukturdienste im Internet (z.B. DNS), Sicherheit
Literatur	<ul style="list-style-type: none">• Peterson L., Davie B., Computer Networks: A Systems Approach, 4th Edition, 2008, Morgan Kaufmann Publishers. (deutsche Übersetzung "Computernetze" erschienen im dPunkt Verlag, Heidelberg, 2008)• Andrew S. Tanenbaum, Computer Networks, 4th Edition, Upper Saddle River, NJ, 2002, Prentice Hall PTR (auch deutsche Übersetzung erschienen bei Pearson Studium, 2003)• Roth J., Prüfungstrainer Rechnernetze, Vieweg+Teubner, 2010
Leistungsnachweis	Schriftliche Prüfung (90 min) Voraussetzungen: erfolgreiche Teilnahme am Praktikum

Rechnersysteme

Studiengang	Bachelor Informatik 2. Studienabschnitt Systeminfrastrukturen
Modul	Rechnersysteme
Modulverantwortliche	Ralf-Ulrich Kern, Axel Hein
Dozent(en)	Axel Hein
Vorkenntnisse	Grundlagen der Informatik, Theoretische Informatik, Programmieren
Arbeitsaufwand	210 Stunden, davon: 95 Stunden Präsenzzeit, 115 Stunden Vor- und Nachbereitung des Lehrstoffs und Prüfungsvorbereitung
Leistungspunkte	7
Semesterwochenstunden	6
Veranstaltungstyp	4 SWS Vorlesung, 2 SWS Übung
Semesterturnus	Wintersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Kenntnisse der Konzepte von Rechnersystemen sind elementares Grundlagenwissen eines Informatikers. Alle Einsatzmöglichkeiten eines Informatikers erfordern die Kenntnis dieser Konzepte sowie konkreter Realisierungen.
Lernziel	Bei Abschluss des Lernprozesses wird der erfolgreiche Studierende in der Lage sein, die wesentlichen Eigenschaften - wie Rechnerstrukturen einschließlich Prozessoren, Peripheriegeräten, Speicherorganisation, Caching, Superskalarität, Pipelining, Multithreading, Cache-Kohärenz und Verbindungsstrukturen, Mikroarchitekturen und Instruktionssatz-Architekturen von Prozessoren, Bewertung der Leistungsfähigkeit und Fehlertoleranz von Rechnersystemen sowie Nutzung von Parallelität - von Computerarchitekturen im Detail zu verstehen, zu analysieren und zu entwerfen.
Schlüsselqualifikation	Fähigkeit zu komplexem Denken, Ausdauer bei Problemlösungen, Abstraktionsvermögen
Lehrinhalte	<ul style="list-style-type: none">• Darstellung wichtiger Rechnerstrukturen, einschließlich Prozessoren, Peripheriegeräten, Speicherorganisation, Caching, Superskalarität, Pipelining, Multithreading, Cache-Kohärenz und Verbindungsstrukturen.• Mikroarchitekturen und Instruktionssatz-Architekturen von Prozessoren.• Bewertung der Leistungsfähigkeit und Fehlertoleranz von Rechnersystemen.• Nutzung von Parallelität.
Literatur	<ul style="list-style-type: none">• Andrew S. Tanenbaum: Structured Computer Organization (deutsch: Computerarchitektur), 2012, Prentice Hall, 6th edition• William Stallings: Computer Organization and Architecture, New Jersey, 2012, Addison Wesley, 9th edition• John L. Hennessy, David A. Patterson: Computer Organization and Design – The Hardware/Software Interface, 2013, Morgan Kaufmann Publishers Inc., 5th edition• John L. Hennessy, David A. Patterson: Computer Architecture – A Quantitative Approach, San Francisco, 2012, Morgan Kaufmann Publishers Inc. / Elsevier, 5th edition

Leistungsnachweis siehe Festlegung in SPO

Hilfsmittel 5(6), 7

Praktikum

Studiengang	Bachelor Informatik 2. Studienabschnitt Praktisches Studiensemester
Modul	Praktikum
Modulverantwortliche	Wolfgang Bremer
Vorkenntnisse	Insbesondere Software Engineering
Arbeitsaufwand	20 Wochen in der Firma
Leistungspunkte	22
Semesterwochenstunden	0
Veranstaltungstyp	20 Wochen praktische Tätigkeit in einem Betrieb
Semesterturnus	Winter- und Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Praktische Erfahrung bei Anwendern und Dienstleistern
Lernziel	Kenntnisse erlangen in der Projektarbeit oder im Servicegeschäft von Informatik- oder Medien-Dienstleistern bzw. in Informatik- oder Medienbereichen. Alternativ sammeln von Erfahrungen in der Projektarbeit in Forschungs- und Entwicklungsbereichen für Informations-, Kommunikations- und Softwaretechnologien.
Schlüsselqualifikation	Projekterfahrung, Zeitmanagement, Sozialkompetenz (Kooperationsfähigkeit, Teamfähigkeit)
Lehrinhalte	Verantwortliche Durchführung von Projekten/Teilprojekten der Softwareentwicklung oder verantwortliche Übernahme von Aufgaben/Teilaufgaben bei der Abwicklung von Informatik-Dienstleistungen
Leistungsnachweis	Ausbildungsbestätigung, Zeugnis

Praxisseminar

Studiengang	Bachelor Informatik 2. Studienabschnitt Praktisches Studiensemester
Modul	Praxisseminar
Modulverantwortliche	Wolfgang Bremer
Dozent(en)	Patricia Brockmann, Ronald Petrlc, Jörg Roth, Joachim Scheja, Heidi Schuhbauer, Matthias Teßmann, Peter Trommler, Bartosz von Rymon Lipinski
Vorkenntnisse	keine
Arbeitsaufwand	60 Stunden, davon: 30 Stunden Präsenzzeit, 30 Stunden Vor- und Nachbereitung der Themen
Leistungspunkte	3
Semesterwochenstunden	2
Veranstaltungstyp	2 SWS Seminar
Semesterturnus	Winter- und Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Grundlagen projektorientierter Arbeit
Lernziel	Erfahrungsaustausch (Fachkenntnisse, Stellung des Praktikanten/der Praktikantin im Unternehmen, Tätigkeiten). Erfolgreiche Präsentation und Dokumentation von Praktikumsergebnissen.
Schlüsselqualifikation	Präsentationskompetenz, Erstellen von Projektberichten
Lehrinhalte	<ul style="list-style-type: none">• Präsentationstechniken• Feedback und Diskussion• Gestaltung von Berichten
Literatur	Seifert, J.W.: Visualisieren, Präsentieren, Moderieren, 20. Auflage, Gabal 2004
Leistungsnachweis	Referate und Praxisbericht

Projektmanagement

Studiengang	Bachelor Informatik 2. Studienabschnitt Praktisches Studiensemester
Modul	Projektmanagement
Modulverantwortliche	Peter Rausch, Thomas Voit
Dozent(en)	Martin Geier, Jörg Meier
Vorkenntnisse	Schulwissen
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenz, 85 Stunden Vor- und Nachbereiten des Lehrstoffs sowie Übungsaufgaben
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	4 SWS Vorlesung mit Übung
Semesterturnus	Wintersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Fähigkeiten in der Organisation und Koordination von zeitlich limitierten Projekten
Lernziel	Methoden der Projektplanung, -steuerung und -kontrolle kennen und anwenden können. Projekte anhand charakteristischer Merkmale definieren und von sonstigen Vorhaben und Formen der Organisationsgestaltung unterscheiden können. Die Werte, Prinzipien und Vorgehensmodelle iterativer und agiler Ansätze kennen und deren spezifische Vor- und Nachteile gegenüber traditionellen Projektmanagement-Ansätzen beurteilen können.
Schlüsselqualifikation	Projektmanagement, Organisationskompetenz, Zeitmanagement
Lehrinhalte	<ul style="list-style-type: none">• Projektplanung, -steuerung und -kontrolle mit Prozessmodellwahl• Meilensteinplanung• Werte, Prinzipien und Vorgehensmodelle des agilen Projektmanagements• Terminplanung mit Netzplantechnik• Kosten- und Aufwandsschätzung• Ressourcenzuordnung• Steuerung und Überwachung• Dokumentation• Überprüfung der Zielerreichung• Lernen aus Projekten
Literatur	<ul style="list-style-type: none">• Hindel, B. u. a.: Basiswissen Projektmanagement, Heidelberg 2004, dpunkt.Verlag• Tumascheit, K. D.: Überleben im Projekt. 10 Projektfallen und wie man sie umgeht, München (Redline) 2014• Heintel, P.; Krainz, E. E.: Projektmanagement. Hierarchiekriese. Systemabwehr. Komplexitätsbewältigung, 5. Auflage, Wiesbaden (Gabler) 2011• Schelle, H.; Ottmann, R.; Pfeiffer, A.: ProjektManager, 3. Auflage, Nürnberg (Deutsche Gesellschaft für Projektmanagement) 2008
Leistungsnachweis	Dieser Leistungsnachweis kann nur praxisbegleitend abgelegt werden. (Klausur 90 Min.)

Hilfsmittel

Zulassungsvoraussetzung

Aktuelle Trends im Einsatz von Projektmanagement-Methoden

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Aktuelle Trends im Einsatz von Projektmanagement-Methoden
Modulverantwortliche	Martin Geier
Dozent(en)	Martin Geier
Vorkenntnisse	Grundlagen in Vorgehensmodellen Grundlagen in Softwareengineering
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenzzeit 40 Stunden Projektarbeit 45 Stunden Vor- und Nachbereitung, Prüfungsvorbereitung
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Vorlesung
Semesterturnus	Winter- bzw. Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Lernziel	Kenntnisse in aktuellen Trends des Software-Projektmanagements und deren Einsatz
Schlüsselqualifikation	Projektmanagement Methoden, Prozessmodelle, Qualitätsmanagement
Lehrinhalte	<p>Projektmanagement ist die zentrale Disziplin, um Projekte auszusteuern und durch Analysen und Abschätzungen Projektmetriken messbar und damit kontrollierbar zu machen. Im Laufe der Zeit haben sich Modelle und Mechanismen als „Best Practices“ etabliert, andere haben an Bedeutung verloren. Die Vorlesung geht auf aktuelle Trends und deren praktischen Einsatz ein und präsentiert entsprechende „Best Practices“.</p> <p>In dieser Lehrveranstaltung werden die folgenden Themen behandelt:</p> <ul style="list-style-type: none">• Einsatz aktueller Projektmanagementmethoden• Aktuelle Verfahrensmodelle und deren Entwicklung• Agile Methoden im Projektmanagement und deren Einsatz (am Beispiel SCRUM)• Vergleich von Projektplanung und Kontrolle in verschiedenen Vorgehensmodellen• Softskills und Mitarbeiterführung / „Human Resource Management“• Aufwandsabschätzungen und deren praktischer Einsatz• Projektbegleitendes Qualitätsmanagement• Reifegradmodelle zur Messung von Prozessqualität
Literatur	<ul style="list-style-type: none">• Hindel, Hörmann, Müller, Schmied: Basiswissen Software-Projektmanagement. dpunkt, 2006• Schwaber: Agile Project Management with Scrum. Microsoft Press, 2004• V-Modell XT. www.v-modell-xt.de
Leistungsnachweis	mündliche Prüfung, Dauer 20 Minuten

Algorithmische Graphentheorie

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Algorithmische Graphentheorie
Modulverantwortliche	Alexander Hufnagel
Dozent(en)	Alexander Hufnagel
Vorkenntnisse	Mathematik 1 und 2, insbesondere Kombinatorik, Lineare Algebra und Rekursionsgleichungen
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenzzeit, 85 Stunden Vor- und Nachbearbeitung des Lehrstoffs und Prüfungsvorbereitung
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Vorlesung mit Übungen
Semesterturnus	Winter- bzw. Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Lernziel	Kenntnis der grundlegenden Begriffe der Graphentheorie sowie der Algorithmen auf Graphen, sowie Übertragung dieser Inhalte auf konkrete Anwendungsbeispiele. Analyse der Laufzeit bzw. Komplexität, Entwickeln von Problemstellungen als Modell der Graphentheorie. Bewertung von Lösungsverfahren für konkrete Projekte.
Lehrinhalte	<p>Graphen zählen zu den wichtigsten Modellen der Informatik. Viele Problemstellungen lassen sich graphentheoretisch beschreiben, beispielsweise</p> <ul style="list-style-type: none">• Ausfallsicherheit von Netzen• Suchstrategien• Finden von kürzesten Wegen• Routenplanung• Zuordnungsprobleme• Maximale Flüsse in Netzwerken <p>Nach einer Einführung in die Theorie und Darstellung von Graphen werden die wichtigsten Algorithmen für Graphentheorie vorgestellt, analysiert und bewertet. Die Methoden werden dann auf konkrete Fragestellungen übertragen. Begleitend zur Vorlesung werden Übungen angeboten.</p>
Literatur	<ul style="list-style-type: none">• Christina Büsing, Graphen- und Netzwerkoptimierung, Spektrum 2010• Shimon Even, Graph Algorithms, 2nd ed., Cambridge 2012• Dieter Jungnickel, Graphs, Networks and Algorithms, 3rd ed., Springer 2007• Sven Oliver Krumke, Hartmut Noltemeier: Graphentheoretische Konzepte und Algorithmen, 3. Auflage, Springer 2012• Angelika Steger, Diskrete Strukturen 1, Springer Verlag 2007• Volker Turau, Algorithmische Graphentheorie, Oldenbourg 2009• T. Cormen et al: Introduction to Algorithms, MIT Press, 2009
Leistungsnachweis	Schriftliche Prüfung (90 min)

Autonome mobile Roboter

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Autonome mobile Roboter
Modulverantwortliche	Jörg Roth
Dozent(en)	Jörg Roth
Vorkenntnisse	Programmieren, Grundlagen der Informatik
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenzzeit, 65 Stunden praktische Studienarbeit, 20 Stunden für Prüfung und Prüfungsvorbereitung
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Vorlesung mit Praktikum
Semesterturnus	Wintersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Kenntnisse über Lösungsstrategien bei der Auswertung von fehlerbehafteten oder unsicheren Informationen, Kenntnisse der Programmierung auf Plattformen mit reduzierten Ressourcen, Fachbezogene Englisch-Kenntnisse.
Lernziel	Bei Abschluss des Lernprozesses wird der erfolgreiche Studierende in der Lage sein, die besonderen Eigenschaften autonomer mobiler Roboter und Verfahren zur autonomen mobilen Robotik zu kennen, Lösungsstrategien zur Erfassung und Interpretation fehlerbehafteter Sensoren anwenden zu können, Planungsmechanismen einsetzen zu können und eine einfache Roboter-Anwendung prototypisch entwickeln zu können.
Schlüsselqualifikation	Fähigkeit zu komplexem Denken, Analyse und Klassifikation von Problemen und Identifikation von Lösungen.
Lehrinhalte	Autonome mobile Roboter können sich selbständig in einer zuerst unbekanntem Umgebung orientieren, bewegen und einfache Aufgaben, beispielsweise Transportaufgaben übernehmen. Bei autonomen mobilen Robotern wird angenommen, dass sich die Sensorerfassung und -auswertung, die Stromversorgung und die notwendige Hard- und Software auf dem Roboter befinden. Die Software-Seite eines autonomen mobilen Roboters muss einerseits die Kartographierung der Umgebung auf der Basis mehrerer Sensoren, andererseits die Planung der Aktionen, insbesondere der Bewegung abdecken. Entsprechende Verfahren müssen dabei mit Ungenauigkeiten bei der Sensorerfassung und Bewegungsausführung umgehen. Darüber ist die Umgebung aus der Sicht eines Roboters komplex, teilweise unbekannt und dynamisch – die Sensoren erfassen dabei meist nur unvollständig alle relevanten Informationen. In der mobilen Robotik wurden verschiedene Verfahren entwickelt, um mit dieser Situation umzugehen. In dieser Veranstaltung wird der Bogen beginnend von der Sensorik bis hin zur Bewegungsplanung von fahrenden und krabbelnden Robotern gespannt. Bestandteil dieser Veranstaltung ist eine Projektarbeit.
Literatur	Kursunterlagen

Leistungsnachweis

Projektarbeit im Umfang von 65 Stunden, mündliche Befragung (20 min), jeweils gewichtet mit 50%

Hilfsmittel

Zulassungsvoraussetzung

Betriebssysteme für einfache Mikrocontroller

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Betriebssysteme für einfache Mikrocontroller
Modulverantwortliche	Thomas Fischer
Dozent(en)	Thomas Fischer
Vorkenntnisse	Grundkenntnisse im Bereich der Rechnerarchitektur und Programmiersprachen
Arbeitsaufwand	75 Stunden, davon: 21 Stunden Präsenz, 14 Stunden zum Lösen der Übungsaufgaben 20 Stunden zum Lesen von Artikeln und Fachbüchern, 20 Stunden zur Nachbereitung und Prüfungsvorbereitung.
Leistungspunkte	2.5
Semesterwochenstunden	2
Veranstaltungstyp	Seminaristischer Unterricht mit Übungen
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs in Deutsch und/oder Englisch (bitte nachfragen)
Beitrag zu den Zielen des Studiums	Vertiefung von Kenntnissen und praktischen Fertigkeiten im Bereich angewandter Softwaretechnologie und Betriebssysteme speziell für eingebettete Systeme (Stichwort IoT)
Lernziel	<ul style="list-style-type: none">• Die AbsolventInnen sind in der Lage Echtzeit-Betriebssysteme für Mikrocontroller zu vergleichen und für eine gegebene Aufgabenstellung auszuwählen, da Sie mit der internen Funktionsweise vertraut sind.• Insbesondere erkennen Sie die Wechselwirkungen zwischen Hardware (speziell Interrupts) und dem eingesetzten Betriebssystem. Aus den vom Betriebssystem angebotenen Mechanismen zur Synchronisation und Koordination können Sie die geeigneten Funktionen auswählen.• Die AbsolventInnen dieser Lehrveranstaltung können unter Einbindung eines einfachen Betriebssystems Anwendungen für Mikrocontroller planen und implementieren.
Schlüsselqualifikation	Fähigkeit zu selbständigem Arbeiten, Fähigkeit zum Verständnis komplexer Systeme Analyse, Interdisziplinarität Fremdsprachenkenntnisse, insbesondere in Englisch -- die LV kann auf Wunsch/ bei Bedarf jederzeit in englischer Sprache angeboten werden! Unterlagen sind fast ausschließlich in englischer Sprache.
Lehrinhalte	Dieser Kurs ist eine Einführung in die Programmierung von Mikrocontrollern mit einfachen (Echtzeit-)Betriebssystemen. Das Thema wird als Überleitung von der Interrupt gesteuerten Programmierung eines Vorder-/Hintergrundsystems zu einer, durch ein Betriebssystem unterstützten, Anwendungssoftware erschlossen: <ul style="list-style-type: none">• Wie laufen Programme in Vorder-/Hintergrundsystemen ab?• Was bestimmt das Zeitverhalten?• Welchen Einfluss haben Interrupts?• Welchen Vorteil bietet die Verwendung eines Betriebssystems?

- Wie unterscheidet sich das Software Design?

Zu diesen Fragen werden folgende Themen vorgetragen bzw. diskutiert:

- Grundlagen und Aufgaben von Betriebssystemen, Geräteverwaltung, Scheduling, Parallelisierung und Prozesssynchronisation.

Im zweiten Teil wird nun der Begriff der Echtzeitsysteme erläutert.

- Welchen Zweck erfüllen Echtzeit-Betriebssysteme?
- Welche Probleme können auftreten, welche Mechanismen sollen diese Probleme verhindern?

Zu diesen Fragen werden folgende Themen vorgetragen bzw. diskutiert:

- Grundlagen der Echtzeitsysteme, Konzepte zu parallelen/ nebenläufigen Prozessen (Prozesskommunikation, Synchronisation, Concurrency).
- Fallstudie des Betriebssystems FreeRTOS und/oder C/OS-III.

In den Übungen werden Rechen- und Programmierbeispiele zum Thema Scheduling und Prozesssynchronisation und Inter Task Kommunikation (Mutex, Queues, Semaphore) an Hand des Betriebssystems FreeRTOS und/oder C/OS-III durchgeführt. Zum Einsatz kommen Entwicklungskits mit dem STM32 Mikrocontroller.

Literatur

Dieser Kurs basiert auf dem Textbuch von

- Barry, Richard: Using the FreeRTOS Real Time Kernel – a Practical Guide, eBook unter www.freertos.org

Weitere Empfohlene Bücher und Artikel sind in Auszügen:

- Kienzle, Friedrich: Programmierung von Echtzeitsystemen, Hanser Verlag 2009
- Labrosse, Jean J.: uC/OS-III – The Real Time Kernel, Micrium Press 2009

Leistungsnachweis

Schriftliche Prüfung (60%) und Übungsaufgaben (40%)

Dauer der schriftlichen Prüfung: 60 Minuten

Die Übungsaufgaben sind eine Projektaufgabe in der die Studierenden in einer Gruppe von 2-4 Personen eine Aufgabenstellung, durch Anwendung der in der Lehrveranstaltung erlernten Fähigkeiten/Kompetenzen, lösen müssen.

Für die Lösung dieser Projektaufgabe steht ein Zeitraum von 2-3 Wochen zur Verfügung.

Der Zeitaufwand pro Studierendem als Beitrag zur Lösung beträgt etwa 20 Stunden.

Die Präsentation und Abgabe erfolgt im Anschluß an die schriftliche Prüfung.

Clean Code in Java-Projekten

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Clean Code in Java-Projekten
Modulverantwortliche	Martin Hock
Dozent(en)	Martin Hock
Vorkenntnisse	Java-Kenntnisse sind von Vorteil.
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenz, 85 Stunden Vor- und Nachbearbeitung, Übungsaufgaben und Prüfungsvorbereitung.
Leistungspunkte	5
Semesterwochenstunden	4
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Vertiefung von Kenntnissen und praktischen Fertigkeiten im Bereich Softwarequalität. Kennenlernen von Tools und Vorgehensweisen aus der Praxis.
Lernziel	Die Studierenden erwerben vertiefte Kenntnisse und Fähigkeiten zur Analyse, Beurteilung und Verbesserung von Software-Qualität. Sie sind in der Lage Prinzipien, Patterns, Techniken und Tools, die zum Schreiben von sauberem Code benötigt werden, anzuwenden.
Schlüsselqualifikation	Fähigkeit zu komplexem Denken, Fähigkeit zu selbständigem Arbeiten, Analyse und Lösung von Problemen.
Lehrinhalte	<p>Die Prüfung von Software ist nicht auf eine bestimmte Phase im Projekt beschränkt. Schon während der Coding-Phase bzw. des System-Build-Prozesses können kritische und schwierig zu findende Softwaredefekte im Quellcode aufgedeckt werden. In der Vorlesung werden die dafür nötigen Verfahren und Tools vorgestellt.</p> <p>Unter anderem werden die folgenden Themen behandelt:</p> <ul style="list-style-type: none">• Übersicht über die Grundlagen der Software-Qualität• Softwaremetriken, Metrikanwendung in der Praxis• Überblick über Prinzipien, Best Practices und Code Smells• Einhaltung und Überprüfung Java Code Conventions• Statische Softwareprüfung, insbesondere Review-Techniken und statische Programmanalyse• Sicherung der Softwarequalität mit Werkzeugen wie SonarQube, PMD, SpotBugs und Checkstyle• Softwaretests mit JUnit• Überprüfen der Testabdeckung (Code Coverage)• Continuous Integration• Design Prinzipien• Design Patterns (GoF)
Literatur	<ul style="list-style-type: none">• Schneider, Kurt: Abenteuer Software Qualität – Grundlagen und Verfahren für Qualitätssicherung und Qualitätsmanagement, dpunkt.verlag, 2007

- Robert, Martin: Clean Code – Refactoring, Patterns, Testen und Techniken für sauberen Code, mitp-Verlag, 2009
- Lilienthal, Carola: Langlebige Software-Architekturen, Dpunkt Verlag, 2015
- Bloch, Joshua: Effective Java – Second Edition, Addison Wesley, 2008
- Roock, Stefan: Refactorings in grossen Softwareprojekten, Dpunkt Verlag, 2004
- Gamma, Erich: Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Professional, 1994
- Robert C. Martin: Agile Software Development: Principles, Patterns and Practices, Prentice Hall, 2003

Leistungsnachweis

Schriftliche Prüfung (90 min)

Hilfsmittel

Zulassungsvoraussetzung

Datenschutz

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Datenschutz
Modulverantwortliche	Ronald Petrlc
Dozent(en)	Ronald Petrlc
Vorkenntnisse	
Arbeitsaufwand	
Leistungspunkte	5
Semesterwochenstunden	4
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	
Lernziel	Die Studierenden haben einen Überblick über die geltenden Gesetze zum Datenschutz. Sie haben die Kompetenz, datenschutzrechtliche Probleme in der betrieblichen Praxis zu erkennen und Handlungsanweisungen daraus abzuleiten.
Schlüsselqualifikation	
Lehrinhalte	<p>Neben einer Einführung in das Datenschutzrecht steht vor allem der technische Datenschutz im Vordergrund. Die Themen werden sehr praxisnah behandelt.</p> <ul style="list-style-type: none">- EU-Datenschutz-Grundverordnung und ePrivacy-Verordnung- Überblick über kryptographische Verfahren- Privacy by Design- Anonymisierung und Pseudonymisierung in der Praxis- Sichere Kommunikation in der Praxis: E-Mail und Messenger- Datenschutz im Web: Tracking, Social Plugins und co.- Identitätsmanagement- Anonymität im Internet- Anonyme Bezahlverfahren / Bitcoin- Datenschutz-Folgenabschätzung
Literatur	<ul style="list-style-type: none">- Petrlc, Sorge: "Datenschutz: Einführung in technischen Datenschutz, Datenschutzrecht und angewandte Kryptographie", Springer-Vieweg, 2017.- Wybitul: „EU-Datenschutz-Grundverordnung im Unternehmen: Praxisleitfaden

Leistungsnachweis

Hilfsmittel

Zulassungsvoraussetzung

Einführung in die Funktionale Programmierung

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Einführung in die Funktionale Programmierung
Modulverantwortliche	Peter Trommler
Dozent(en)	Peter Trommler
Vorkenntnisse	Programmieren I und II, Mathematik I, Algorithmen und Datenstrukturen
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenz, 45 Stunden selbständige Übungen am Rechner, 40 Stunden zur Nachbereitung und Prüfungsvorbereitung.
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Vorlesung mit Praktikum
Semesterturnus	Wintersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Spezifikation komplexer Systeme und deren Implementierung mit funktionalen Programmen
Lernziel	Programmieren im funktionalen Programmierparadigma, Verbesserung der Analysefähigkeit und Programmierkompetenz auch in objektorientierten Sprachen
Schlüsselqualifikation	Abstraktionsvermögen, Durchhaltevermögen, logisches Schließen und komplexes Denken
Lehrinhalte	<p>Am Beispiel der Programmiersprache Haskell werden die Prinzipien des Funktionalen Programmierens betrachtet und in praktischen Übungen im Labor eingeübt.</p> <p>Folgende Themenbereiche werden behandelt:</p> <ul style="list-style-type: none">• Typen, Klassen, Parametrische Polymorphie• Funktionen, Definition, Komposition• Rekursion, Funktionen höherer Ordnung, Kombinatoren• Reine Funktionen und Nebeneffekte, Monaden• Auswertungsstrategien: strikte und faule Auswertung, unendliche Listen
Literatur	<ul style="list-style-type: none">• Graham Hutton, Programming in Haskell, Cambridge University Press, Cambridge, England, 2. Auflage, 2016.• Brian O'Sullivan, John Goerzen, and Don Stewart, Real World Haskell, O'Reilly, Sebastopol, California, U.S.A., 2008.• Richard Bird, Pearls of Functional Algorithm Design, Cambridge University Press, Cambridge, England, 2010.• Richard Bird, Introduction to Functional Programming using Haskell, Prentice Hall Europe, Harlow, Essex, England, 2. Auflage, 1998.• Miran Lipovaca, Learn You a Haskell for Great Good, no starch press, San Francisco, California, U.S.A., 2011.
Leistungsnachweis	Mündliche Prüfung (Dauer: 20 Minuten)

Hilfsmittel

Zulassungsvoraussetzung

Einführung in Operations-Research

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Einführung in Operations-Research
Modulverantwortliche	Claudia Hirschmann
Dozent(en)	Claudia Hirschmann
Vorkenntnisse	Grundlagen der Mathematik, Mathematik-Vorlesungen.
Arbeitsaufwand	Umfang 2 SWS, und häusliche Arbeiten
Leistungspunkte	2.5
Semesterwochenstunden	2
Veranstaltungstyp	Die Veranstaltung wird auf vier Tagesblöcke im Semester verteilt. Zwischen den Tagesblöcken ist Arbeit in Übungsgruppen einzuplanen und auszuführen. Die Tagesblöcke sind interaktiv gestaltet (Mitarbeiten der Teilnehmenden).
Semesterturnus	Winter- bzw. Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Grundlagenwissen, Modelle, Methoden, Herangehensweisen zum Operations-Research in Theorie und Praxis, vertieft durch praxisorientierte und Transfer-Übungen. Mathematische und Operations-Research Grundlagen z.B. der linearen Programmierung und anderer Methoden zur Entscheidungsunterstützung. Analytische Fähigkeiten bzgl. Problemstellungen im Hinblick auf Operations- Research. Befähigung zum Transfer von Modellen aus dem Operations- Research auf verschiedene Problemstellungen aus der Informatik, der Geschäftsplanung und dem Prozessmanagement. Selbständige Erarbeitung von (kleineren) Teilgebieten des FWPF, Problemlösung in Arbeitsgruppen, Präsentation der Arbeitsergebnisse.
Lernziel	Einführung, Kenntnis ausgewählter Begriffe und Vorgehensweise aus Teildisziplinen des Operations- Research, und Anwendungsbeispiele aus dem Operations-Research. Unter „Operations-Research“ wird die Modell-gestützte Vorbereitung von Entscheidungen zur Gestaltung und Lenkung sozio-technischer Systeme verstanden. Es ist geprägt durch die Zusammenarbeit von Angewandter Mathematik, Wirtschaftswissenschaften, Informatik und einigen Ingenieurwissenschaften.
Schlüsselqualifikation	Fähigkeit zur grundlegenden mathematischen Modellbildung, zur Analyse und zur Klassifikation von Fragestellungen aus der Praxis u.a. aus der Geschäftsplanung, betriebswirtschaftlichen Fragestellungen und dem Prozessmanagement als Grundlage für die weiteren Lösungsschritte. Fähigkeit zum Verständnis komplexer Probleme und deren Lösungsansätze aus dem Operations- Research, Transfer von Mathematischen Modellen auf verwandte Fragestellungen, Interdisziplinarität, Bearbeitung interdisziplinärer Fragestellungen, Teamarbeit, Kommunikations- und Teamfähigkeiten, Präsentationsfähigkeit, selbständiges Problemlösen und Problemlösen in Arbeitsgruppen.
Lehrinhalte	Die Veranstaltung führt in die Arbeitsweise und Teilgebiete des „Operations-Research“, d.h. der Optimierung von technisch-wirtschaftlichen Problemstellungen unter Beachtung von Nebenbedingungen (wie z.B. Ressourcenbeschränkungen), ein. Ein Einstieg in Operations- Research ist die Betrachtung von Problemstellungen aus

der Praxis, die sich mittels linearer Gleichungen und Ungleichungen beschreiben lassen. Ein in der Vorlesung behandeltes Teilgebiet ist die lineare Optimierung. Neben den reinen Lehrinhalten zu den Teilgebieten des Operations-Research nehmen Anwendung und praktische Beispiele und Übungen einen breiten Rahmen ein.

Literatur

- R. E. Burkard, U. T. Zimmermann: „Einführung in die Mathematische Optimierung“, Springer Verlag; - W. Domschke: Einführung in Operations Research, Springer Verlag; - B. Werners: Grundlagen des Operations Research, Mit Aufgaben und Lösungen, Springer Verlag

Leistungsnachweis

Referat mit Handout (30 Min.) (50%), schriftliche Befragung (60 Min.) (50%); Aufgrund des Veranstaltungsdesigns herrscht Anwesenheitspflicht.

Embedded Systems

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Embedded Systems
Modulverantwortliche	Friedhelm Stappert
Dozent(en)	Friedhelm Stappert
Vorkenntnisse	Grundlagen Betriebssysteme, Grundlagen Rechnerarchitektur, Programmierkenntnisse
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenzzeit, 85 Stunden zur Vor- und Nachbereitung des Lehrstoffs und Prüfungsvorbereitung
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Vorlesung mit Praktikum
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Vermittlung von Kenntnissen in einem wesentlichen Bereich der angewandten Informatik
Lernziel	<ul style="list-style-type: none">• Kenntniss der speziellen Eigenschaften und Anforderungen von eingebetteten Systemen• Hardware-Architekturen für eingebettete Systeme verstehen und bewerten• Einfache, hardwarenahe Programme für Mikrocontroller erstellen können• Software-Architekturen für eingebettete Systeme kennen und beurteilen• Modellierungsparadigmen für eingebettete Systeme kennen und anwenden• Fähigkeit zum Entwurf eingebetteter Systeme
Schlüsselqualifikation	Fähigkeit zu komplexem, analytischem Denken; Fähigkeit zu Analyse und Verständnis komplexer Systeme; Ausdauer bei Problemlösungen
Lehrinhalte	<p>Ein Embedded System ist eine HW/SW-Einheit, die über Sensoren und Aktoren in ein Gesamtsystem "eingebettet" ist und spezielle Überwachungs-, und Regelungsaufgaben übernimmt.</p> <p>Embedded Systems finden sich in vielen Bereichen des täglichen Lebens, von der Waschmaschinensteuerung bis zum Airbag-Steuergerät im Automobil. Täglich kommen wir mit Dutzenden solcher Systeme -- mehr oder weniger unbewusst -- in Berührung.</p> <p>Dieser Kurs vermittelt grundlegende Kenntnisse zum Entwurf von eingebetteten Systemen:</p> <ul style="list-style-type: none">• Besondere Eigenschaften, Anwendungsbereiche und Anforderungen von eingebetteten Systemen• Hardware-Architekturen, Microcontroller• Software für eingebettete Systeme• Design / Modellierung eingebetteter Systeme• Echtzeitsysteme

Literatur

Peter Marwedel. *Eingebettete Systeme*. Springer 2007.
Berns, Schürmann, Trapp. *Eingebettete Systeme; Systemgrundlagen und Entwicklung eingebetteter Software*. Vieweg + Teubner 2010.
Jean J. Labrosse. *Embedded Software*. Newnes 2008.
Peter Scholz. *Softwareentwicklung eingebetteter Systeme*. Springer 2005.

Leistungsnachweis

Schriftliche Prüfung (90 Minuten). Aktive Teilnahme am Praktikum ist Voraussetzung für die Klausur

Graphical User Interface Design and Information Visualization

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Graphical User Interface Design and Information Visualization
Modulverantwortliche	Axel Platz
Dozent(en)	Axel Platz
Vorkenntnisse	Grundsätzliche Kenntnisse von interaktiven Systemen und Software-Ergonomie
Arbeitsaufwand	150 Stunden, davon: 60 Stunden Präsenz und 90 Stunden Vor- und Nachbereitung
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Vorlesung und praxisorientierte Übungen
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Die Erstellung von fortgeschrittenen Benutzeroberflächen erfordert informationstechnische, gestalterische und ergonomie-relevante Kenntnisse und Fähigkeiten. Diese Lehrveranstaltung ergänzt die gestalterischen Fähigkeiten.
Lernziel	In der Lehrveranstaltung soll die Fähigkeit zur Konzeption und Gestaltung graphischer Bedienoberflächen in Theorie und Praxis vermittelt und durch praxisorientierte Übungen vertieft werden.
Schlüsselqualifikation	Kreatives Problemlösen, praktische Erfahrung, Interdisziplinarität
Lehrinhalte	Usability / Ästhetik / User Experience, Mensch-Maschine Interaktion, Kommunikationstheorie, Nutzungskontexterhebung (Analyse), Konzeption und visuelle Gestaltung (Synthese), Wahrnehmungsphysiologie und -psychologie (u.a. Gestaltgesetze, optische Täuschungen), Theorie des Bildes / Bildtypen und Bildfunktionen (insbesondere Strukturprojektionen), Visuelle Sprache (Expressivität von Formen und Farben), Anschauliches Denken (Theorie anschaulichen Denkens bei Rudolf Arnheim)
Literatur	Florin, Alexander: User Interface Design Ware, Colin: Information Visualization Arnheim, Rudolf: Visual Thinking, University of California Press, 1969
Leistungsnachweis	1/3 Referat (20 Minuten) und 2/3 Studienarbeit (Konzeption eines User Interface)

Grundlagen Content-Management-Systeme

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Grundlagen Content-Management-Systeme
Modulverantwortliche	Alexander Kröner
Dozent(en)	Alexander Kröner
Vorkenntnisse	Erforderlich <ul style="list-style-type: none">• Fach "Programmieren II"• Fach "Web-Programmierung" (oder äquivalente Fähigkeiten in HTML, CSS, JavaScript, PHP) Empfohlen <ul style="list-style-type: none">• Grundlagen SQL (z.B. Fach "Datenbanken")
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenzzeit 85 Stunden Vor- und Nachbereitung, Prüfungsvorbereitung
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Vorlesung, Projektarbeit
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Die Entwicklung von Erweiterungen für Web-Content-Management-Systeme stärkt einerseits praktische Fähigkeiten im Bereich der Web-Programmierung, und eröffnet zugleich einen Zugang zu den diesen populären Anwendungen zugrundeliegenden Softwarearchitekturen.
Lernziel	Kenntnis charakteristischer Merkmale und der Struktur von Content-Management-Systemen (CMS). Anwendung dieses Wissens, um ein Anwendungsszenario hinsichtlich funktionaler Anforderungen an ein CMS zu analysieren. Fähigkeit, Software-Komponenten zur Umsetzung der gewünschten Funktionalität zu konzipieren, implementieren und in eine bestehende Softwarearchitektur zu integrieren.
Schlüsselqualifikation	Implementierung von Content Management Systemen, Softwarearchitekturen Web-basierter Systeme, Softwareentwicklung in einer heterogenen Technologielandschaft.
Lehrinhalte	Content-Management-Systeme (CMS) sind in einer zunehmend durch digitale Daten bestimmten Medienwelt ein zentrales Werkzeug zur Steuerung von Erstellung, Organisation, und Präsentation von Inhalten. Schwerpunkt der Vorlesung bilden Web-basierte CMS. Behandelt u.a.:

- Grundlagen: Features, Klassifizierung, Auswahlkriterien, Architekturkonzepte, Content-Lifecycle, Asset-Management
- CMS erweitern: Frontend und Backend erweitern, Plugins entwickeln und einbinden, Personalisierung, Internationalisierung und Lokalisierung, Integration strukturierter Daten
- Schnittstellen: Import und Export von Content, Content Syndication, Client-side Scripting

Praktische Arbeiten erfolgen mehrheitlich auf Basis von PHP sowie dem CMS WordPress.

Literatur

- B. Boiko (2004). Content Management Bible. 2te Auflage. John Wiley & Sons.

Leistungsnachweis

Seminarleistung bestehend aus:

- 10% Referat (10 Min.)
- 90% Studienarbeit (Softwareanwendung entsprechend 75 Stunden Entwicklungsaufwand)

Grundlagen der Computerspielentwicklung

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Grundlagen der Computerspielentwicklung
Modulverantwortliche	Bartosz von Rymon Lipinski
Dozent(en)	Bartosz von Rymon Lipinski
Vorkenntnisse	Programmieren I und II, Algorithmen und Datenstrukturen, Computergraphik. Vorteilhaft: Gestaltungs- und Medienlehre I und II, Softwarearchitektur, Software Engineering.
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenz, 85 Stunden Vor- und Nachbearbeitung, Übungsaufgaben und Prüfungsvorbereitung.
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Vorlesung, Seminar, Projektarbeit
Semesterturnus	Wintersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Vermittlung von Kenntnissen aus der praktischen und angewandten Informatik im Anwendungsfeld Computerspiele. Zugang zum interdisziplinären Gebiet, in dem Softwareentwicklung und Gestaltung interaktiver Systeme zusammenwirken. Befähigung komplexe computergraphische Anwendungen zu verstehen, konzipieren und herzustellen.
Lernziel	Kompetenz die technischen sowie kreativen elementaren Rahmenbedingungen bei der Entwicklung einer interaktiven virtuellen Welt zu erläutern und zu bewerten. Grundlegende Kenntnisse hinsichtlich der Planung und Konzeption eines Computerspiels. Fähigkeiten zur Auswahl und Beurteilung geeigneter Softwarewerkzeuge und darauf basierend zur prototypischen Umsetzung.
Schlüsselqualifikation	Analyse und Klassifikation von Problemen, kreatives Problemlösen, selbständiges Arbeiten, Kommunikations- und Teamfähigkeit, Interdisziplinarität.
Lehrinhalte	1. Einführung und Geschichte der Computerspiele 2. Planung einer Computerspielproduktion 3. Grundlagen des (technischen) „Game Designs“ 4. Technische Architektur von Computerspielen - Einordnung 5. Überblick und Einordnung bzgl. „Game Engines“ 6. Planung, Konzeption und Umsetzung eines Spielprototypen
Literatur	<ul style="list-style-type: none">• Fullerton, T. J.: Game Design Workshop, Apple Academic Press Inc., 2013• Schell, J.: The Art of Game Design - A Book of Lenses, Morgan Kaufmann, 2014• Rabin, S.: Introduction to Game Development, Cengage Delmar, 2009• Gregory, J.: Game Engine Architecture, Taylor & Francis Ltd., 2014• Nystrom, R.: Game Programming Patterns, Genever Benning, 2014

- McShaffry, M., Graham, D.: Game Coding Complete, Cengage Learning, 2012
- Novak, J.: Game Development Essentials, Delmar Cengage Learning, 2011

Leistungsnachweis

Praktische Studienarbeit (Kommentiertes Software-Produkt, entsprechend 85 Stunden Entwicklungsaufwand).

Grundlagen der Software Security

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Grundlagen der Software Security
Modulverantwortliche	Hans Löhr
Dozent(en)	Hans Löhr
Vorkenntnisse	Programmierkenntnisse (C/C++), Grundlagen Rechnerarchitektu
Arbeitsaufwand	150 Stunden, davon: ca. 60 Stunden Präsenz, 90 Stunden Heimarbeit (Vortragsvorbereitung, Bericht schreiben, Vor- und Nachbereitung)
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Seminar
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Vertiefung von Kenntnissen und praktischen Fertigkeiten im Bereich angewandter Softwaretechnologie; Software-Security-Kenntnisse;
Lernziel	<ul style="list-style-type: none">• Fachkompetenz: Die Studierenden verstehen die Ursachen von Software-Schwachstellen sowie Konzepte der sicheren Software-Entwicklung und die Grundprinzipien für einen sicheren Entwicklungszyklus.• Methodenkompetenz: Die Studierenden erlernen die Fähigkeit, Software-Schwachstellen und Problemstellungen zu analysieren, Sicherheitstechniken auszuwählen und kritisch zu evaluieren; darüberhinaus üben sie die Fähigkeit, einen wissenschaftlichen Vortrag auszuarbeiten und zu halten sowie eine wissenschaftliche Ausarbeitung dazu anzufertigen.• Soziale Kompetenz: Organisation, Planung, Präsentationstechnik.
Schlüsselqualifikation	Analyse komplexer Sachverhalte, Präsentationstechnik, Anfertigung einer wissenschaftlichen Ausarbeitung
Lehrinhalte	Die Studierenden erhalten in dem Kurs einen vertiefenden Einblick in verschiedene Themenfelder aus dem Bereich Software Security: <ul style="list-style-type: none">• Software Schwachstellen und Angriffstechniken• Verschiedene Arten von Schwachstellen und Angriffen, Kategorisierungen, Common Weakness Enumeration (CWE). Beispiele:<ul style="list-style-type: none">◦ buffer overflows◦ integer overflows◦ injection attacks◦ return-oriented programming• Sicherheitsaspekte verschiedener Programmiersprachen, z.B.:<ul style="list-style-type: none">◦ Java◦ C#◦ Rust• Software Security Testing Tools

- Software Fuzzing
- Statische Security-Analyse und Analyse-Tools
 - Syntaktische Checks
 - Graph-basierte tools
 - Verifikation und formale Methoden
- Sicherheit im Software-Entwicklungszyklus
- Sicherheitsarchitekturen und Software-Härtung

Literatur

themenspezifische Fachartikel;
Lehrbuch u.a.: Gary McGraw, John Viega, "Building Secure Software"

Leistungsnachweis

Seminarleistung und Gewichtung:

- Referat mit Demo bzw. praktischer Übung (ca. 45 Minuten; 50%)
- schriftliche Ausarbeitung (ca. 15 Seiten; 40%)
- aktive Teilnahme an der Diskussion (10%)

Hilfsmittel

Zulassungsvoraussetzung

Grundlagen des maschinellen Lernens

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Grundlagen des maschinellen Lernens
Modulverantwortliche	Tobias Bocklet
Dozent(en)	Tobias Bocklet
Vorkenntnisse	Keine
Arbeitsaufwand	150h (60h Präsenzzeit, 90h Eigenstudium)
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	4 SWS seminaristischer Unterricht mit Vorlesung und Übungen
Semesterturnus	Winter- bzw. Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	
Lernziel	Die Studierenden kennen und verstehen die theoretischen und algorithmischen Grundlagen des maschinellen Lernens. Sie können Probleme aus dem Bereich des Maschinellen Lernens erkennen und selbständig lösen.
Schlüsselqualifikation	Fähigkeit zu komplexem Denken, Fähigkeit zu selbständigem Arbeiten, Analyse und Klassifikation von Problemen, kreatives Problemlösen
Lehrinhalte	<ul style="list-style-type: none">• Grundlegende Prinzipien und Theorien des Maschinellen Lernens• Einführung der wichtigsten algorithmischen Konzepte• Lernen aus Daten• Praktische Anwendungen
Literatur	<ul style="list-style-type: none">• Duda, R and Hart, P and Stork, D: Pattern Classification, 2. Auflage, 2001, Wiley, NY• Geron A: HAnds-On Machine Learning with Scikit-Learn, Keras & TensorFlow, 2. Auflage, 2019, O'Reilly Media• Niemann,H: <i>Klassifikation von Mustern</i>. 2. Überarbeitete Auflage, 2003• Goodfellow, I and Bengio,Y and Courville, A: <i>Deep Learning</i>. 2016
Leistungsnachweis	Schriftliche Prüfung (90min) am Veranstaltungsende
Hilfsmittel	
Zulassungsvoraussetzung	

Grundlagen des Softwaretests

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Grundlagen des Softwaretests
Modulverantwortliche	Ricarda Dormeyer
Dozent(en)	Ricarda Dormeyer
Vorkenntnisse	Zu Test & Qualitätssicherung: keine. Programmierkenntnisse sind wünschenswert, ebenso Grundkenntnisse zu Vorgehensmodellen und Software Engineering.
Arbeitsaufwand	150h, davon: 65h Präsenz, 85h Nachbereitung des Lehrstoffs und Prüfungsvorbereitung
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Blockveranstaltung
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Weil Softwarefehler schwerwiegende Folgen haben können, muss jede Software vor ihrem Einsatz getestet werden. Deshalb sind Grundkenntnisse über systematisches Testen von Software für Informatiker wichtig.
Lernziel	Erlernen grundlegender Konzepte, Begriffe und Methoden zu Test und Qualitätssicherung von Software
Schlüsselqualifikation	Test und Qualitätssicherung von Software (Komponenten, Systemen, Systemlandschaften), Testprozesse und ihre Integration in Softwareentwicklungsprozesse, Testmethoden, Testarten, Testrollen, Testwerkzeuge
Lehrinhalte	<p>Die Veranstaltung deckt die Inhalte des ISTQB Certified Tester Foundation Level ab. Dabei handelt es sich um ein seit vielen Jahren weltweit anerkanntes und etabliertes Aus- und Weiterbildungsschema für Softwaretester.</p> <p>Themen sind insbesondere:</p> <ul style="list-style-type: none">• Motivation des Softwaretests• Grundbegriffe• Fundamentaler Testprozess• Teststufen und Testarten• Statische Qualitätssicherung• Testbarkeit von Requirements• Testmethoden (Black Box, White Box, erfahrungsbasiert)• Konzeption und Planung von Tests• Wirtschaftlichkeit des Testens• Testrollen• Kategorien von Testwerkzeugen.

Literatur

Spillner, Linz: Basiswissen Softwaretest. dpunkt Verlag, 6. Auflage

Ergänzend/vertiefend:

Spillner, Roßner, Winter, Linz: Praxiswissen Softwaretest - Testmanagement. dpunkt Verlag, 4. Auflage

Roßner, Brandes, Götz, Winter: Basiswissen Modellbasierter Test. dpunkt Verlag.

Crispin, Gregory: Agile Testing. Addison-Wesley

Seidl, Baumgartner, Bucsics: Basiswissen Testautomatisierung. dpunkt Verlag

Leistungsnachweis

Mündliche Prüfung im Anschluss an die Blockveranstaltung (20 min)

Grundlagen von C++ für C#-Entwickler

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Grundlagen von C++ für C#-Entwickler
Modulverantwortliche	Bartosz von Rymon Lipinski
Dozent(en)	Bartosz von Rymon Lipinski
Vorkenntnisse	Vorkenntnisse in objektorientierter Programmierung mit C# (erfolgreich abgeschlossenes Programmieren II)
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenz, 85 Stunden Vor- und Nachbearbeitung, Übungsaufgaben und Prüfungsvorbereitung
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Seminar, Projektarbeit
Semesterturnus	Wintersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Vermittlung von Kenntnissen aus der praktischen und angewandten Informatik. Befähigung implementatorische Herausforderungen einzuordnen und praxisnahe Lösungen umzusetzen.
Lernziel	Kompetenz die konzeptionellen und technischen Rahmenbedingungen in der modernen Programmierung zu erläutern und zu bewerten. Fähigkeit zur Umsetzung eines objektorientierten Systems (mit der Programmiersprache C++). Fähigkeit hinsichtlich des Transfers zwischen verschiedenen objektorientierten Programmiersprachen (C++ und C#).
Schlüsselqualifikation	Praktisches Problemlösen, selbständiges Arbeiten, Kommunikations- und Teamfähigkeit.
Lehrinhalte	<ul style="list-style-type: none">• Modernes C++: Einführung und Abgrenzung zu C#• Einrichtung der Entwicklungsumgebung• Prozedurale Programmierung (Variablen, Kontrollstrukturen, Funktionen, Ein-/Ausgabe, Zeiger/Referenzen, Präprozessor)• Objektorientierung (Klassen/Structs, Vererbung und Polymorphie, inkl. virtuelle und abstrakte Methoden)• Weiterführendes (Namensräume, Überladen von Operatoren, Ausnahmebehandlung, Grundlagen von Templates)• Überblick über die C++ Standardbibliothek (String- und Container-Klassen, Smart pointers, mathematische Berechnungen, Algorithmen)• Eventuell erste Schritte mit dem Qt Framework
Literatur	<ul style="list-style-type: none">• Wolf, J.: Grundkurs C++ - C++-Programmierung verständlich erklärt, Rheinwerk Computing, 2016• Kalista, H.: C++ für Spieleprogrammierer, Carl Hanser Verlag GmbH & Co. KG, 2016• Stroustrup, B.: Die C++-Programmiersprache - Aktuell zu C++11, Carl Hanser Verlag GmbH & Co. KG, 2015• Meyers, S.: Effective Modern C++ - 42 Specific Ways to Improve Your Use of C++11

and C++14, O'Reilly Media, 2014

Leistungsnachweis

Praktische Studienarbeit (Kommentiertes Software-Produkt, entsprechend 60 Stunden
Entwicklungsaufwand).

Zulassungsvoraussetzung

In-Memory Computing am Beispiel von SAP HANA

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	In-Memory Computing am Beispiel von SAP HANA
Modulverantwortliche	Rainer Weber
Dozent(en)	Rainer Weber
Vorkenntnisse	Kenntnisse über Datenbanksysteme, insbesondere von SQL. Programmierkenntnisse in einer Programmiersprache (nicht notwendigerweise der im Kurs verwendeten). SAP-Erfahrung ist nicht erforderlich.
Arbeitsaufwand	75 Stunden, davon: 30 Präsenzzeit, 45 Stunden Vor- und Nachbereitung des Lehrstoffs
Leistungspunkte	2.5
Semesterwochenstunden	2
Veranstaltungstyp	Lehrveranstaltung mit Vorlesungs- und Übungsanteilen
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs in Deutsch und/oder Englisch (bitte nachfragen)
Beitrag zu den Zielen des Studiums	Die erworbenen Kenntnisse und Methoden sind nützlich für Aufgaben im Bereich „Business Intelligence“ und Anwendungsentwicklung.
Lernziel	Kenntnis des Potentials von In-Memory Computing für Unternehmenssoftware. Fähigkeit, In-Memory-Anwendungen durch Modellierung und durch Softwareentwicklung zu erstellen.
Schlüsselqualifikation	Fähigkeit zum Verständnis großer komplexer Systeme, Lernen am Beispiel, Transferfähigkeit, algorithmisches Denken, Teamfähigkeit
Lehrinhalte	<p>Der Kurs befasst sich mit einem frischen Thema: In-Memory Computing, sowohl im Allgemeinen wie auch speziell am Beispiel des recht neuen Systems SAP HANA.</p> <p>Bei In-Memory-Computing wird die vollständige Datenbank im Hauptspeicher gehalten statt auf einer Magnetplatte. Es wird ermöglicht durch immer größere und billigere Hauptspeicher und die parallele Bearbeitung durch Multicore-Prozessoren. Dadurch lassen sich Antwortzeiten komplexerer analytischer Anwendungen vom Minutenbereich in den Sekundenbereich bringen („sub-second response time“). Bestimmte Auswertungen sowie neuartige Anwendungen werden dadurch erst möglich.</p> <p>Der Kurs befasst sich vor allem mit der Architektur von Anwendungen, die das Potential von In-Memory Computing nutzen. Ist traditionell die Datenbank der Performanzengpass eines betrieblichen Anwendungssystems, so läuft die Anwendungslogik in einem In-Memory System umgekehrt möglichst nahe an der Datenbank.</p> <p>In dem Kurs werden die Konzepte und die Architektur von In-Memory Systemen geschildert (z.B. die spaltenorientierte Speicherung). Der wesentliche Teil ist allerdings</p>

die Nutzung von In-Memory Systemen, insbesondere an praktischen Übungen mit SAP HANA. Zum einen werden analytische Anwendungen durch Modellierung (ohne Programmierung) erstellt, die Auswertungen geschehen mit Business-Intelligence-Werkzeugen. Hierbei erfolgt auch eine Einführung in die Begriffe und Methoden von Business Intelligence und Data Warehousing. Zum anderen werden Anwendungen programmiert. In der Lehrveranstaltung wird dies im wesentlichen die Programmierung von Stored Procedures mit der Programmiersprache SQLScript sein. Ergänzende Programmiertechnologien werden kürzer angesprochen: REST-basierte Web-Services und serverseitiges JavaScript für die schmale Applikationsserverschicht und clientseitiges JavaScript und SAPUI5-Technologie für die breitere Client-seitige Präsentationsschicht.

Geplant ist auch, auf Beispiele für neuartige Anwendungen einzugehen.

Literatur

H. Plattner, G. Zeier: In-Memory Data Management: Technology and Applications. 2. Auflage. Springer, Berlin Heidelberg New York, 2012
B. Berg, P. Silvia: Einführung in SAP HANA. 2. Auflage. Galileo Press, Bonn, 2013
T. Schneider, E. Westenberg, H. Gahm: ABAP Entwicklung für SAP HANA. Galileo Press, Bonn, 2013

Leistungsnachweis

Mündliche Prüfung (Gruppenprüfung, bei zwei Personen 30 Minuten)

Lean Startup

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Lean Startup
Modulverantwortliche	Korbinian Riedhammer
Dozent(en)	Korbinian Riedhammer
Vorkenntnisse	Keine
Arbeitsaufwand	150h (60h Präsenzzeit, 90h Eigenstudium)
Leistungspunkte	5
Semesterwochenstunden	4
Semesterturnus	Winter- bzw. Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Systematische Erarbeitung von technischen wie wirtschaftlichen Anforderungen
Lernziel	Die Studierenden erlernen die Lean Innovation und Lean Startup Methoden und können diese auf Problemstellungen im IT Umfeld anwenden. Sie können Innovationen aus technischer und unternehmerischer Sicht bewerten und mit geeigneten Werkzeugen prototypische Lösungen realisieren.
Schlüsselqualifikation	Problemanalyse, unternehmerisches Denken, agile Methoden
Lehrinhalte	<p>Lean Innovation ist eine Methode zur Steigerung der Effizienz durch das frühe und konsequente Einbeziehen von Benutzer- bzw. Kundenfeedback. Eine Vorgehensweise zur strukturierten Generierung und Entwicklung von Ideen ist hierbei Design Thinking, bei dem Lösungen iterativ und in enger Abstimmung mit den Benutzern erfolgt. Lean Startup ist eine wissenschaftliche Herangehensweise zur Ausarbeitung von Geschäftsideen welche auf der strukturierten Definition und experimentellen Validierung oder Widerlegung von Hypothesen beruht. Dazu wird Schritt für Schritt ein Business Model Canvas (BMC) erstellt, bei dem das Wertversprechen (Value Proposition) im Vordergrund steht. Die folgenden Methoden und Aktivitäten werden erarbeitet und angewendet:</p> <ul style="list-style-type: none">• Lean Innovation und Design Thinking• Lean Startup• Business Model Canvas• Customer Development• Zielgerichtete Interviews: Gesprächsführung und -beobachtung• Pitchen: Ideen erfolgreich präsentieren
Literatur	<ul style="list-style-type: none">• Steve Blank. The Startup Owner's Manual: The Step-By-Step Guide for Building a Great Company• Alexander Osterwalder, Yves Pigneur. Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers• Eric Ries. The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses• Frank Rimalovski, Giff Constable, Tom Fishburne. Talking to Humans: Success starts with understanding your customers

Leistungsnachweis	Semesterbegleitende Projektarbeit mit 12 wöchentlichen Milestones (je max. 10 Punkte) und abschließender Präsentation (max. 20 Punkte). Die Note wird aus der Gesamtpunktzahl (max. 140) bestimmt.
Hilfsmittel	2
Zulassungsvoraussetzung	Keine

Logik

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Logik
Modulverantwortliche	Yvonne Stry, Peter Trommler
Dozent(en)	Yvonne Stry
Vorkenntnisse	Mathematik, Theoretische Informatik
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenzzeit 30 Stunden Vor- und Nachbereiten des Vorlesungsstoffes 55 Stunden Vorbereiten der Präsentation
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Vorlesung und Seminar
Semesterturnus	Wintersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Theoretischer Hintergrund
Lernziel	<ul style="list-style-type: none">• Einführung in die Aussagenlogik und in die Prädikatenlogik• Studieren der Entwicklung der Logik und ihrer zahlreichen Zweige
Schlüsselqualifikation	Präsentationskompetenz
Lehrinhalte	Vorlesung: Aussagenlogik, Prädikatenlogik Vortragsthemen: Geschichte der Logik, spezielle Logiken, Methoden des logischen Schließens, formale Methoden der Softwareentwicklung, Künstliche Intelligenz, automatisches Beweisen etc.
Literatur	Uwe Schöning, Logik für Informatiker, Spektrum Akademischer Verlag
Leistungsnachweis	Seminarvortrag 25 min

Praxis des Software-Tests eingebetteter Systeme

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Praxis des Software-Tests eingebetteter Systeme
Modulverantwortliche	Kristian Trenkel
Dozent(en)	Kristian Trenkel
Vorkenntnisse	Basiswissen zu Eingebetteten Systemen
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Blockveranstaltung
Semesterturnus	Wintersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Die fortschreitende Vernetzung und die Integration von Systemen, vor allem von eingebetteten Systemen, stellt nicht nur Anforderungen an die Entwicklung dieser Systeme. Der Test dieser Systeme ist ein immer größerer Bestandteil der Arbeit von Ingenieuren. Das in dieser Veranstaltung vermittelte Wissen zum Software-Tests eingebetteter Systeme erhöht die Problemlösungskompetenz der Studierenden und verbesserte Ihre Möglichkeiten zur interdisziplinären Zusammenarbeit.
Lernziel	Die Studierenden können den Software-Test für eingebettete Systeme praktisch anwenden. Dabei liegt der Fokus auf den verstehen Funktionaler Tests und dem anwenden des Hardware In The Loop (HIL)-Testverfahrens. Dabei sollen die Studierenden die Abläufe von Testaktivitäten und deren Aufwände innerhalb des Entwicklungsprozesses eingebetteter Systeme erkennen und bewerten können. Den Studierenden sollen die Möglichkeiten und Vorteile des HIL-Testverfahrens bewerten können. Des Weiteren sollen sie auch die Grenzen des Testverfahrens beurteilen und bewerten. Anhand von Beispielen wird den Studierenden das Anwender der Testfahren und Testsysteme praktisch verdeutlicht. Es werden die Begrifflichkeiten entsprechend des ISTQB Glossars wie in der Lehrveranstaltung „Grundlagen des Softwaretests“ verwendet.
Lehrinhalte	<ul style="list-style-type: none">• Motivation für den Software-Test mit Fokus auf Eingebettete Systeme• Einordnung des Tests in den Entwicklungsprozess• Wiederholung fundamentaler Testprozesse• Wiederholung Teststufen und Testarten• Einführung des HIL-Testverfahrens (Technische Entwicklung, Einsatzbereiche, Vorgehen, ...)• Einordnung des HIL-Testverfahrens in die Teststufen• Erläuterung des Aufbaues von typischem HIL-Testsystemen (inklusive der Echtzeitsysteme verschiedener Hersteller)• Erläuterung des Aufgabenverteilung innerhalb des Testsystems (z.B. Echtzeit-Modell, Ablaufsteuerung des Tests)• Darstellung der Tool-Umgebungen für HIL-Testsysteme• Erläuterungen der Testautomatisierung• Darstellung verschiedener Testfälle und ihrer Umsetzung am Beispiel eines

Automobil-Steuergerätes

- Umsetzung eines beispielhaften Testfalles von den Anforderungen über die Testspezifikation und die Testimplementierung bis hin zu den Testergebnissen
- Erläuterung der Schnittstellen zu anderen Testverfahren
- Erläuterung der Grenzen des HIL-Testverfahrens
- Durchführung praktischer Beispiele

Literatur

Grünfelder, Stephan.: "Software-Test für Embedded Systems - Ein Praxishandbuch für Entwickler, Tester und technische Projektleiter", dpunkt.verlag, 2013

Leistungsnachweis

Schriftliche Prüfung - Dauer: 90 min

Programmieren von Mikrocontrollern

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Programmieren von Mikrocontrollern
Modulverantwortliche	Thomas Fischer
Dozent(en)	Thomas Fischer
Vorkenntnisse	Grundlagen der C-Programmierung Rechnerarchitektur Grundlagen
Arbeitsaufwand	150 Stunden, davon: 64 Stunden Präsenz, 46 Stunden zum Lösen der Übungsaufgaben 20 Stunden zum Lesen von Artikeln und Fachbüchern, 20 Stunden zur Nachbereitung und Prüfungsvorbereitung.
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Seminaristischer Unterricht mit Übungen
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs in Deutsch und/oder Englisch (bitte nachfragen)
Beitrag zu den Zielen des Studiums	Vertiefung von Kenntnissen und praktischen Fertigkeiten im Bereich Funktionsweise von Computer-Systemen speziell für eingebettete Systeme (Stichwort IoT)
Lernziel	<ul style="list-style-type: none">• Die Studierenden können einfache kombinatorische und sequentielle Systeme entwerfen und anwenden. Sie sind imstande einfache Probleme durch Auswahl eines Mikrocontrollers und Erstellung des Programms in C zu lösen. Sie können im Entwurfsprozess verschiedene Lösungswege vergleichen und die geeigneten Peripheriekomponenten auswählen.• Für den Entwurf von Anwenderprogrammen für eingebettete Systeme (Internet of Things) sind die Studierenden in der Lage die Möglichkeiten der verwendeten Hardware zu analysieren und die dafür notwendigen Konfigurationsparameter abzuleiten.
Schlüsselqualifikation	Fähigkeit zu selbständigem Arbeiten, Fähigkeit zum Verständnis komplexer Systeme Analyse und Klassifikation von Problemen und Identifikation von Lösungen, Teamfähigkeit, Interdisziplinarität Fremdsprachenkenntnisse, insbesondere in Englisch -- die LV kann auf Wunsch/ bei Bedarf jederzeit in englischer Sprache angeboten werden! Unterlagen sind fast ausschließlich in englischer Sprache.
Lehrinhalte	Grundlagen des Aufbaus der Hardware und der internen Baugruppen eines Datenverarbeitungssystems (MC8). Interne Abläufe bei der Ausführung von Befehlen eines Programmes. Unterschiede Mikroprozessor - Mikrocontroller - System on Chip. Grundlagen der Schnittstellen und Peripheriekomponenten (interne und externe Kommunikation). Timer, Interrupts, USART, PS/2 Tastatur, LCD Display. Lösen einer Problemstellung (STM32 - ARM Cortex M3). Erstellen von C-Programmen zur Lösung verschiedener Aufgabenstellungen

angepasst an die verwendete Hardware.
Überleitung von der Interrupt gesteuerten Programmierung eines
Vorder-/Hintergrundsystems zu einer, durch ein Betriebssystem unterstützten
modularen Programmierung.
Zum Einsatz kommen Entwicklungskits mit dem STM32 Mikrocontroller.

Literatur

Herstellerdokumentation von ARM und ST (www.arm.com ;www.st.com)

Leistungsnachweis

Schriftliche Prüfung (60 min, 50%) und Projektaufgabe (20 Std., 50%)

Programmierung und Technologie betrieblicher Standardsoftware (SAP-Praktikum)

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Programmierung und Technologie betrieblicher Standardsoftware (SAP-Praktikum)
Modulverantwortliche	Rainer Weber
Dozent(en)	Rainer Weber
Vorkenntnisse	Programmierung (mindestens eine Programmiersprache), Datenbanksysteme, insbesondere SQL
Arbeitsaufwand	150 Stunden, davon: 65 Präsenz, 85 Stunden Vor- und Nachbereitung des Lehrstoffs und Prüfungsvorbereitung
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Üblicherweise Blockkurs mit intensivem Anteil von praktischen Übungen. Im Wintersemester 2017/2018 nicht als Blockkurs, sondern während der regulären Vorlesungszeit.
Semesterturnus	Wintersemester
Unterrichtssprache	Kurs in Deutsch und/oder Englisch (bitte nachfragen)
Beitrag zu den Zielen des Studiums	Die erworbenen Kenntnisse und Methoden befähigen zur Übernahme von Aufgaben in der Anwendungsentwicklung und im Informationsmanagement (Entwicklung, Erweiterung und Anpassung von betrieblicher Standardsoftware).
Lernziel	Die spezifischen Methoden und Techniken zur Erstellung betrieblicher Standardsoftware kennenlernen Fähigkeit, kleinere Anwendungsprogramme in der Programmiersprache ABAP erstellen zu können
Schlüsselqualifikation	Fähigkeit zum Verständnis großer komplexer Systeme, Lernen am Beispiel, Transferfähigkeit, algorithmisches Denken, Teamfähigkeit
Lehrinhalte	<ul style="list-style-type: none">• Programmierung betrieblicher Standardsoftware am Beispiel von SAP-Software in der Programmiersprache ABAP<ul style="list-style-type: none">◦ Grundlegende ABAP-Sprachelemente◦ Datenbankzugriffe, Data Dictionary◦ ABAP Objects◦ Web Dynpro für ABAP◦ Business Add-ins• Ausgewählte Aspekte der Technologie von Unternehmenssoftware
Literatur	H. Keller, S. Krüger: ABAP Objects. Galileo Press, 3. Auflage, 2006 R. Weber: Technologie von Unternehmenssoftware. Berlin – Heidelberg (Springer), 2012
Leistungsnachweis	Mündliche Prüfung (Gruppenprüfung, bei zwei Personen 30 Minuten)

Requirements Engineering

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Requirements Engineering
Modulverantwortliche	Ramin Tavakoli Kolagari
Dozent(en)	Ramin Tavakoli Kolagari
Vorkenntnisse	Software Engineering
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenzzeit, 85 Stunden Vor- und Nachbereitung des Lehrstoffs und Prüfungsvorbereitung
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	2 SWS Vorlesung, 2 SWS Übung
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs in Deutsch und/oder Englisch (bitte nachfragen)
Beitrag zu den Zielen des Studiums	<p>In der frühen Phase der Softwareentwicklung wird über den Projekterfolg großer Softwareprojekte entschieden: Werden falsche, widersprüchliche oder unerfüllbare Anforderungen an das Projekt gestellt, so können diese im besten Fall unter Einsatz hoher Kosten in nachfolgenden Phasen angepasst und damit das Projekt insgesamt durchgeführt werden; im in der Realität häufig anzutreffenden schlimmsten Fall wird das Projekt insgesamt nicht zum Abschluss gebracht.</p> <p>Die Analysephase nimmt damit eine Schlüsselrolle ein, die durch ihren spezifischen („weichen“) Charakter im Studiengang Informatik besondere Aufmerksamkeit verdient.</p>
Lernziel	<p>Fähigkeit zur Erhebung, Spezifikation und Inspektion von Anforderungen.</p> <p>Gute Kenntnis der UML Analysemodelle, ausgewählter Techniken und Methoden des modernen Requirements Engineering und Requirements Management.</p> <p>Vorbereitung auf die Prüfung „Certified Requirements Engineer“ (http://www.certified-re.de/) und bei Interesse: anschließende Durchführung der Prüfung (kostenpflichtig!, für Studierende die Hälfte der aktuellen Zertifizierungsgebühr von circa 220,- EUR, Zertifizierer sind derzeit der iSQI https://www.isqi.org/ und der TÜV SÜD http://www.tuev-sued.de/home_de).</p>
Schlüsselqualifikation	<p>Fähigkeit zur Verbalisierung von Gedanken (gemäß Friedrich Dürrenmatt: „Die Arbeit an der Sprache ist eine Arbeit am Gedanken.“).</p> <p>Fertigkeit zur Anwendung von UML Modellen in der Analyse sowie von Techniken und Methoden zur Erhebung, Beschreibung und Prüfung von Anforderungen.</p>
Lehrinhalte	Dieser Kurs vermittelt grundlegendes Verständnis von Methoden und Prinzipien des Requirements Engineering. Dazu gehören Kenntnisse über die Analysephase im Allgemeinen sowie über

- Techniken und Methoden zur Erhebung, Spezifikation und Inspektion von Anforderungen,
- UML Analysemodelle,
- Analysis Patterns,
- Problem Frames und
- weitergehende Kenntnisse für das erfolgreiche Absolvieren der Prüfung „Certified Requirements Engineer“, Foundation Level, <http://www.certified-re.de/>

im Besonderen. Die Inhalte werden in Übungen sowie in Literaturarbeit vertieft.

Literatur

- C. Rupp: "Requirements-Engineering und -Management: Das Handbuch für Anforderungen in jeder Situation", Hanser Verlag, 7. Auflage, 2020.
- K. Pohl, C. Rupp: " Basiswissen Requirements Engineering Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering Foundation Level ", dpunkt.verlag, 5. Auflage 2021.
- C. Rupp, S. Queins: "UML 2 glasklar: Praxiswissen für die UML-Modellierung", Hanser Verlag, 4. Auflage 2012.
- K. Pohl: "Requirements Engineering: Fundamentals, Principles, and Techniques", Springer Verlag, 2010.

Leistungsnachweis

Befragung (20 Min., 50%) sowie ein Referat (20 Min. + interaktive Übung und Diskussion, 50%).

Software-Ergonomie in der Medizintechnik

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Software-Ergonomie in der Medizintechnik
Modulverantwortliche	Martin Herget
Dozent(en)	Martin Herget
Vorkenntnisse	Praktische Erfahrung in der Entwicklung von Anwendungen mit einer Benutzerschnittstelle
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenzzeit, 85 Stunden Durchführung des Projekts und Vorbereitung der Präsentationen
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Vorlesung und Projektarbeit
Semesterturnus	Wintersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Die Prinzipien leicht verständlicher und gut benutzbarer Software sind relevant für alle Arten von Software mit einer Benutzerschnittstelle. Besonders wichtig ist ihre Beachtung für Softwaresysteme in einem Umfeld mit Gefährdungspotential. Bewusstsein und Befähigung zur Herstellung ergonomischer Softwaresysteme ist eine wichtige Kompetenz.
Lernziel	Kenntnis der Konzepte für die Gestaltung und der Kriterien für die Beurteilung und Validierung ergonomischer graphischer Dialogsysteme. Fähigkeit zur systematischen und Benutzer-orientierten Vorgehensweise nach allgemein anerkannten Prinzipien. Fähigkeit, benutzerfreundliche Softwaresysteme zu entwerfen.
Schlüsselqualifikation	Fähigkeit zu komplexem Denken, Fähigkeit zu selbständigem Arbeiten, Analyse und Klassifikation von Problemen, kreatives Problemlösen, Präsentationskompetenz
Lehrinhalte	Einführung in die Ergonomie, Definition und Einordnung der Ergonomie, Ergonomie im Entwicklungsprozess, Software-Ergonomie, Vorgehensmodelle, User-Centered-Design, Methoden der Aufgabenanalyse und Modellierung, Metaphern und konzeptuelle Modelle als Mittel der UI-Gestaltung, Regeln zur Layout- und Dialoggestaltung, Überblick Style Guides und ihre Bedeutung für die Medizintechnik, relevante Ergonomie-Normen, Evaluationsverfahren und Kriterien zur Bewertung von Benutzungsoberflächen, Usability Testing, Praxisbeispiele aus der Medizintechnik.
Literatur	<ul style="list-style-type: none">• Usability Engineering, Jacob Nielsen, 1993• Don't make me think, Steve Krug, 2006• Florian Sarodnick, Methoden der Usability Evaluierung, Huber, 2006• Hoa Loranger, Jakob Nielsen: Web Usability, Addison-Wesley, 2008• The Usability Engineering Lifecycle, Deborah J. Mayhew, Morgan Kaufmann, 1999
Leistungsnachweis	<ul style="list-style-type: none">• schriftliche Klausur (90 Minuten Dauer)• Ergebnisdokumentation der Projektarbeit (Umfang ca. 10 Seiten)

Software-Zuverlässigkeit: Theorie und Praxis

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Software-Zuverlässigkeit: Theorie und Praxis
Modulverantwortliche	Harald Stieber
Vorkenntnisse	Mathematik III (Statistik)
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenzzeit, 85 Stunden Vor- und Nachbearbeitung des Lehrstoffs und Prüfungsvorbereitung
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Vorlesung
Semesterturnus	Winter- bzw. Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Testen ist ein wesentlicher Teil des Software-Lebenszyklus. Software-Zuverlässigkeit ist ein wichtiger Aspekt der Software-Qualität.
Lernziel	Kenntnis der Grundprinzipien des Software-Tests. Fähigkeit Fehlerdaten zu sammeln und mit geeigneten mathematischen Modellen die Software-Zuverlässigkeit abzuschätzen und zu verbessern.
Schlüsselqualifikation	Logisches Denkvermögen
Lehrinhalte	<ul style="list-style-type: none">• Testmethodik• Benutzungsprofile• Testen nach Benutzungsprofil• statistische Modelle zur Vorhersage der Software-Zuverlässigkeit• Methoden zur Überprüfung der Vorhersagegenauigkeit• Optimierung des Testprozesses
Literatur	<ul style="list-style-type: none">• M. Lyu (Herausgeber): Handbook of Software Reliability Engineering, McGraw-Hill• A. Mathur: Foundations of Software Testing, Addison-Wesley• Musa, Iannino, Okumoto: Software Reliability - Measurement, Prediction, Application, McGraw-Hill (1987)
Leistungsnachweis	Schriftlich, 60 Minuten
Hilfsmittel	
Zulassungsvoraussetzung	

Technical Computing

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Technical Computing
Modulverantwortliche	Oliver Kluge
Dozent(en)	Oliver Kluge
Vorkenntnisse	Programmiererfahrung
Arbeitsaufwand	150 Stunden, davon 60 Stunden Präsenzzeit und 90 Stunden Vor- und Nachbereitung.
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Vorlesung und Übung
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Vertiefung der Theorie des Technical Computing und Anwendung in der Praxis
Lernziel	Ob bei der statistischen Analyse von Messwerten, der Berechnung des Amplitudenspektrums oder der Filterung im Zeit- und Frequenzbereich, Werkzeuge wie MatLab & Co bieten dem Entwickler jede Menge Unterstützung an. Aber wie funktionieren sie? Der Kurs befähigt die Studierenden die zugrundeliegenden Algorithmen zu analysieren, deren Leistungsfähigkeit zu beurteilen und eigene Lösungen zu entwickeln. Weitere Informationen: https://www.technical-computing.de
Schlüsselqualifikation	Analytisches Denken, Abstraktionsvermögen, Fähigkeit zum Verständnis komplexer Systeme
Lehrinhalte	Algorithmen der digitalen Signalverarbeitung (DFT, IDFT, FFT, STFT, Filter im Zeit- und Frequenzbereich), Algorithmen der statistischen Analyse (Korrelation, Regression, grafische Verfahren)
Literatur	https://www.technical-computing.de/references.html
Leistungsnachweis	Mündliche Prüfung (30 Minuten)
Hilfsmittel	
Zulassungsvoraussetzung	

Text Analytics

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Text Analytics
Modulverantwortliche	Jens Albrecht
Dozent(en)	Jens Albrecht
Vorkenntnisse	Wünschenswert: Datenbanken, Programmieren 1 und 2
Arbeitsaufwand	150 Stunden, davon: 50 Stunden Präsenzzeit 100 Stunden Vor- und Nachbereitung der Vorlesung, Abschlussprojekt und Abschlussprüfung
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Seminaristischer Unterricht mit praktischen Übungen. Die notwendigen theoretischen Konzepte werden in der Vorlesung gemeinsam erarbeitet und im Übungsteil exemplarisch in umgesetzt bzw. unter Verwendung existierender Frameworks angewandt. Es wird mit Python gearbeitet.
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums Lernziel	Nach der Teilnahme an diesem Modul sind die Studierenden in der Lage <ul style="list-style-type: none">• wichtige Begriffe im Bereich des Natural Language Processing zu definieren• überwachte und unüberwachte maschinelle Lernverfahren für die Analyse unstrukturierter, natürlichsprachlicher Texte zu erklären, zu bewerten und einzusetzen• Textdaten mit Python systematisch aufzubereiten, zu visualisieren und zu analysieren• eigenständig Lösungsansätze für analytische Fragestellungen zu natürlichsprachlichen Texten zu entwickeln• Machine-Learning-Prozesse in Python umzusetzen
Schlüsselqualifikation	Fähigkeit zu selbständigem Lernen, komplexem Denken, Abstraktions- und Transferfähigkeit, analytische Kompetenz, Teamfähigkeit
Lehrinhalte	Ein Großteil der digital verfügbaren Information liegt in Form unstrukturierter Texte vor, und diese Datenmenge wächst täglich. Aufgrund der Eigenheiten natürlicher Sprache erfordert die automatisierte Erschließung von Web-Inhalten, Nutzer-Kommentaren, Emails oder digitalisierten Gesprächsprotokollen besondere Verfahren und Algorithmen. Diese kommen beispielsweise in Suchmaschinen, Spam-Filtern oder bei der Verbrechensbekämpfung zum Einsatz. Ziel der Veranstaltung ist es, die

Studierenden mit grundlegenden Methoden zum Auffinden von Information (Information Retrieval), zur automatisierten Analyse von Textinhalten (Text Mining) und zur Extraktion von Wissen (Knowledge Acquisition) vertraut zu machen. Im Einzelnen wird auf folgende Aspekte eingegangen:

- Grundlagen des Natural Language Processing (NLP)
 - Vagheit in der Sprache
 - Aufbereitung von Texten
 - Part-of-Speech-Tagging
 - Named Entity Recognition
- Information Retrieval
 - Vektorraum-Modell
 - Ähnlichkeit von Dokumenten
 - Arbeitsweise von Suchmaschinen
- Text Mining und Machine Learning auf Texten
 - Topic-Analyse und Clustering
 - Klassifikation
 - Meinungs- und Stimmungsanalyse
 - Kontextuelle Ähnlichkeitsanalyse mit Word Embeddings

Literatur

Albrecht, Jens; Ramachandran, Sidharth; Winkler, Christian: Blueprints for Text Analytics Using Python. O'Reilly, 2020.
<https://learning.oreilly.com/library/view/-/9781492074076/?ar>

Lane, Hobson; Cole, Howard; Hapke, Hannes Max: Natural Language Processing in Action. Manning, 2019.
<https://learning.oreilly.com/library/view/-/9781617294631/?ar>

Bengfort, Benjamin ; Bilbro, Rebecca ; Ojeda, Tony: Applied Text Analysis with Python. O'Reilly, 2018.

Bird, Steven ; Klein, Ewan ; Loper, Edward: Natural Language Processing with Python. O'Reilly, 2009. Online frei verfügbar unter <http://www.nltk.org/book/>

Henrich, Andreas: Information Retrieval 1. Universität Bamberg, Lehrstuhl für Medieninformatik, 2008. Online frei verfügbar unter <https://www.uni-bamberg.de/minf/ir1-buch/>

Manning, Christopher D. ; Raghavan, Prabhakar ; Schütze, Hinrich: An Introduction to Information Retrieval. Cambridge University Press, 2009. Online frei verfügbar unter <https://nlp.stanford.edu/IR-book/>

Raschka, Sebastian: Python Machine Learning. Packt Publishing, 2015. Online im Campusnetz verfügbar unter <http://proquest.tech.safaribooksonline.de/9781783555130> (auf Deutsch bei MITP Verlag erschienen, aber nicht online verfügbar)

VanderPlas, Jake: Python Data Science Handbook. O'Reilly, 2016. Komplett als Jupyter-Notebooks verfügbar unter <https://jakevdp.github.io/PythonDataScienceHandbook>

Leistungsnachweis

Studienarbeit in Form eines Abschlussprojekts mit Vortrag und mündliche Befragung (15 min), Gewichtung jeweils 50%.

Hilfsmittel

Zulassungsvoraussetzung

Virtuelle Maschinen

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Virtuelle Maschinen
Modulverantwortliche	Ralf-Ulrich Kern
Dozent(en)	Ralf-Ulrich Kern
Vorkenntnisse	Rechnersysteme; unbedingt erforderlich: bestandene Prüfung in "Betriebssysteme"
Arbeitsaufwand	75 Stunden, davon: 30 Stunden Präsenzzeit, 45 Stunden für Vor- und Nachbereitung des Lehrstoffs, Vorbereitung des Seminarvortrags und schriftliche Ausarbeitung.
Leistungspunkte	2.5
Semesterwochenstunden	2
Veranstaltungstyp	Seminar
Semesterturnus	Wintersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Vermittlung von Spezialkenntnissen im Fachgebiet Betriebssysteme, die für vielfältige Gebiete der Informatik nützlich sind.
Lernziel	Analyse der architektonischen Voraussetzungen, Vergleich verschiedener Typen und Implementierungsformen von virtuellen Maschinen und Fähigkeit, das Einsatzpotential von virtuellen Maschinen zu beurteilen.
Schlüsselqualifikation	Analyse und Klassifikation von Problemen, kreatives Problemlösen, Präsentationskompetenz
Lehrinhalte	<p>Mit Virtualisierung im hier verwendeten Sinne ist die Replizierung (Vervielfältigung) realer Hardware gemeint. D. h., aus einem Rechner werden mittels einer Software-Schicht virtuell (d. h. scheinbar) mehrere Rechner ("virtuelle Maschinen") gemacht, und in jeder dieser virtuellen Maschinen kann zur selben Zeit ein eigenes Betriebssystem ablaufen, isoliert von allen anderen.</p> <p>Diese Technik ist seit ca. 1965 bekannt, als sie zunächst für IBM-Großrechner entwickelt wurde, und sie ist aus verschiedenen Gründen zu neuer Blüte gekommen, da es dafür viele Anwendungsmöglichkeiten sowohl auf dem Desktop als auch im Server-Bereich sowie in eingebetteten Systemen gibt.</p> <p>In diesem Seminar sollen abseits des Marketingrummels die technischen Grundlagen aus den Bereichen Prozessorarchitektur und Betriebssysteme erarbeitet werden. Ausgangspunkt ist die immer noch aktuelle Arbeit von Popek und Goldberg, in der Bedingungen für die effiziente Virtualisierbarkeit einer Prozessorarchitektur angegeben wurden. Da diese Bedingungen in der klassischen x86-Architektur von Intel und AMD nicht erfüllt waren, mussten für die Implementierung virtueller Maschinen Umgehungsmöglichkeiten gefunden werden, die auch behandelt werden. Inzwischen haben Intel und AMD jeweils eigene Virtualisierungserweiterungen der x86-Architektur entwickelt, die den Popek/Goldberg-Bedingungen genügen und die die</p>

Implementierung effizienter virtueller Maschinen vereinfachen.

Nach einer kurzen Vorlesungseinführung in das Thema werden in Vorträgen die historische Entwicklung, die technischen Grundlagen und auch eine Auswahl aus der laufend anwachsenden Menge von aktuellen Virtualisierungssystemen beleuchtet, wobei bei letzteren der Schwerpunkt auf die Varianten der methodischen Vorgehensweise im Zusammenhang mit dem jeweiligen Einsatzgebiet gelegt wird.

Stichwörter zum Inhalt:

Virtuelle Maschine, Virtualisierungssystem, Popek/Goldberg-Bedingungen

Literatur

- POPEK, G. J. ; GOLDBERG, R. P.: Formal Requirements for Virtualizable Third Generation Architectures.
In: Communications of the ACM (1974).
- ROBIN, J. S. ; IRVINE, C. E.: Analysis of the Intel Pentium's ability to Support a Secure Virtual Machine Monitor.
In: Proceedings of the 9th Usenix Symposium (2000).
- ADAMS, K. ; AGESEN, O. : A comparison of software and hardware techniques for x86 virtualization.
In: ASPLOS-XII: Proceedings of the 12th international conference on Architectural support for programming languages and operating systems. New York, NY, USA : ACM Press, 2006.

Leistungsnachweis

Referat (max. 45 min) und Studienarbeit (10 - 15 Seiten, Ausarbeitung des Referats); Gewichtung je 50 %. Anwesenheitspflicht während der Seminarvorträge

Visualisierung

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Visualisierung
Modulverantwortliche	Matthias Teßmann
Dozent(en)	Matthias Teßmann
Vorkenntnisse	Grundlagen der Informatik, Algorithmen und Datenstrukturen, Computergraphik, Programmierkenntnisse (C++, Python), Mathematik (insb. Lineare Algebra, Differential- & Integralrechnung)
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenz 85 Stunden Vor- und Nachbereitung des Lehrstoffes sowie Übungsaufgaben
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Vorlesung und Seminar
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs in Deutsch und/oder Englisch (bitte nachfragen)
Beitrag zu den Zielen des Studiums	Die Visualisierung von Daten aus technisch-wissenschaftlichen Experimenten oder Simulationen zum Zwecke des Erkenntnisgewinnes ist ein wichtiges Anwendungsgebiet der Informatik. Kenntnis der Methoden zur Verarbeitung und Darstellung dieser Daten können auf vielfältige Weise eingesetzt werden.
Lernziel	Analyse und Vergleich der grundlegenden Algorithmen zur Visualisierung von Daten aus technisch-wissenschaftlichen Experimenten. Bewertung der Anwendbarkeit verschiedener Verfahren in Bezug auf spezifische Fragestellungen. Entwicklung von Strategien zur Bearbeitung neuer Problemstellungen.
Schlüsselqualifikation	Analyse und Klassifikation von Problemen, kreatives Problemlösen, selbständiges Arbeiten, Fähigkeit zu komplexem Denken, Fähigkeit zum Verständnis komplexer Systeme und Algorithmen, Lernen am Beispiel, Transferfähigkeit
Lehrinhalte	Die Visualisierung beschäftigt sich mit der visuellen Aufbereitung der (oft großen) Datenmengen aus technisch-wissenschaftlichen Experimenten oder Simulationen zum Zweck des tieferen Verständnisses und der einfacheren Präsentation komplexer Phänomene. Die Vorlesung gibt eine Einführung in die grundlegenden Algorithmen und Datenstrukturen, sowie einen Überblick über die verfügbaren Softwarewerkzeuge und verbreiteten Dateiformate. Behandelt werden u.a. folgende Themen: <ul style="list-style-type: none">• Visualisierungsszenarien• Gitterstrukturen und Interpolation• Verfahren für 2D Skalar- und Vektorfelder• Verfahren für 3D Skalar- und Vektorfelder

- Verfahren für multivariate Daten
- Volumenvisualisierung mit Isoflächen
- Direktes Volume-Rendering

Literatur

- W. Schroeder and K. Martin, "The Visualization Toolkit", Kitware Inc. 2004
- M. Ward, G.G. Grinstein and D. Keim, "Interactive Data Visualization: Foundations, Techniques, and Applications", Taylor & Francis, 2010
- C.D. Hansen and C.R. Johnson, "Visualization Handbook", Academic Press, 2004

Leistungsnachweis

Schriftliche Prüfung (90 Minuten, 100%)

Hilfsmittel

5(5), 7

Zulassungsvoraussetzung

Voice-over-IP

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Voice-over-IP
Modulverantwortliche	Matthias Teßmann
Dozent(en)	Matthias Teßmann
Vorkenntnisse	Grundlagen der Informatik, Algorithmen und Datenstrukturen, Programmieren I & II (zusätzliche Kenntnisse in C und C++ wären von Vorteil), Rechnerkommunikation
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenz, 85 Stunden Vor- und Nachbereiten des Lehrstoffs sowie Übungsaufgaben
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Seminar mit Praktikum
Semesterturnus	Wintersemester
Unterrichtssprache	Kurs in Deutsch und/oder Englisch (bitte nachfragen)
Beitrag zu den Zielen des Studiums	Voice-over-IP ist ein wichtiges Anwendungsgebiet der Informatik und mittlerweile die wichtigste Technologie für globale Sprachkommunikation
Lernziel	Identifikation der wesentlichen Einzelkomponenten eines komplexen Softwareprodukts am konkreten Beispiel einer Voice-over-IP Anwendung. Gegenüberstellung verschiedener Implementierungsansätze für die einzelnen Komponenten (z.B. Audio-Codecs, Jitter-Buffer, ...), sowie die Analyse der daraus resultierenden Auswirkungen auf das Gesamtsystem. Aufzeigen der Schwierigkeiten und Herausforderungen, die sich für den Betrieb eines Voice-over-IP-Systems in einem offenen Netzwerk ergeben. Fähigkeit zur Bewertung der Rolle freier Standards und Protokolle, die eine weitreichende Interoperabilität verschiedener Implementierungen erst ermöglichen. Sammeln praktischer Erfahrung durch die Implementierung einer konkreten Voice-over-IP Anwendung.
Schlüsselqualifikation	Analyse und Klassifikation von Problemen, kreatives Problemlösen, selbständiges Arbeiten, Fähigkeit zu komplexem Denken, Fähigkeit zum Verständnis großer komplexer Systeme, Lernen am Beispiel, Transferfähigkeit
Lehrinhalte	<ul style="list-style-type: none">• Einführung in Telekommunikationssysteme und -netze• Audiosignalverarbeitung: Sprache, Töne & Musik• Sprach- und Audiokodierung<ul style="list-style-type: none">◦ POTS, HD-Voice, Full-HD Voice• Standards und Standardisierungsprozesse• IP-Protokollfamilie: IP, TCP, UDP, SCTP• Das Real-Time Transport Protokoll (RTP)• IP-basierte Echtzeitkommunikation<ul style="list-style-type: none">◦ Herausforderungen: Delay, Jitter, Packet Loss, QoS

Literatur

- RFCs 3261, 3550, 3640, 4040, 4961 u.a. (<http://www.ietf.org/rfc.html>)
- R. Swale, D. Collins, "Carrier Grade Voice Over IP", McGraw-Hill Professional, 3rd Ed., 2013
- C. Perkins, "RTP: Audio and Video for the Internet", Adison Wesley, 2012
- D. Minoli, E. Minoli, "Delivering Voice over IP Networks", John Wiley & Sons, 2nd Ed., 2002

Leistungsnachweis

Referat (45 Minuten, 60%) und praktische Studienarbeit (kommentiertes Software-Produkt entsprechend 25 Stunden Entwicklungsaufwand, 40%). Während der Präsentationen besteht Anwesenheitspflicht.

Hilfsmittel

Zulassungsvoraussetzung

Web Application Security

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Web Application Security
Modulverantwortliche	Peter Trommler, Ralf Reinhardt
Dozent(en)	Ralf Reinhardt, Peter Trommler
Vorkenntnisse	<ul style="list-style-type: none">• Grundkenntnisse der Programmierung (Vorlesung 'Programmieren 1 und 2', 1. und 2. Semester)• Grundkenntnisse über Computer-Netzwerke und verteilte Anwendungen (Vorlesung 'Rechnerkommunikation', 4. Semester)
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenz, 45 Stunden zum Lesen von Artikeln, 40 Stunden zur Nachbereitung und Prüfungsvorbereitung.
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Seminaristischer Unterricht mit Übungen
Semesterturnus	Wintersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Vertiefung von Kenntnissen und praktischen Fertigkeiten im Bereich angewandter Softwaretechnologie, speziell im Bereich sicherer Software Systeme
Lernziel	Fähigkeit zur praktischen, sicheren Verwendung von Web-Technologien, Analysieren von Architekturen von Webanwendungen, Analysieren von Angriffsszenarien, praktische Durchführung einiger konkreter Angriffe, Fähigkeit zur Anwendung von Techniken zur manuellen und automatisierten Erkennung von Sicherheitslücken in Webanwendungen und Bewertung der Resultate
Schlüsselqualifikation	Erkennen und Beseitigung von Sicherheitsrisiken in Webanwendungen
Lehrinhalte	Gefahren und Angriffe: (mit Übungen) <ul style="list-style-type: none">• OWASP Top-10• Injection Angriffe• Validierung von Eingaben• Cross-Site Angriffe Gegenmaßnahmen <ul style="list-style-type: none">• Web Application Firewalls (WAF)• Automatisierte Erkennung von Sicherheitslücken
Literatur	<ul style="list-style-type: none">• Sverre Huseby, Innocent Code: A Security Wake-Up Call for Web Programmers (auch deutsch d-punkt Verlag)• The Open Web Application Security Project: http://www.owasp.org• Web Application Exploits and Defenses: http://google-gruyere.appspot.com/

Leistungsnachweis

Mündliche Prüfung (Dauer: 20 Minuten)

Zulassungsvoraussetzung

Datenbanken II in der Wirtschaftsinformatik

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer wählbar außerhalb Schwerpunkt
Modul	Datenbanken II in der Wirtschaftsinformatik
Modulverantwortliche	Alfred Holl
Dozent(en)	Alfred Holl
Vorkenntnisse	Datenbanken
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenzzeit, 85 Stunden für Vorbereitung von Referat und Ausarbeitung
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Seminar (Bachelor)
Semesterturnus	Winter- bzw. Sommersemester
Unterrichtssprache	Kurs in Deutsch und Englisch
Beitrag zu den Zielen des Studiums	Vertiefte Kenntnisse über Datenbanken spielen in vielen Berufsfeldern von Absolventen informatikorientierter Studiengänge eine wichtige Rolle.
Lernziel	Kenntnisse über Ziele und Methodiken von Datenbanken Fähigkeit, unterschiedliche Typen von Datenbanken in verschiedenen Anwendungsbereichen in der Wirtschaftsinformatik einzusetzen
Schlüsselqualifikation	Fähigkeit zur Analyse komplexer Probleme Umgang mit wissenschaftlicher Literatur technisch-wissenschaftliches Schreiben Präsentationsfähigkeiten
Lehrinhalte	Eine Auswahl der folgenden Themen: <ul style="list-style-type: none">• Datenbanken im Internet• temporale Datenbanken• Multimedia-Datenbanken• sqlite• Datenbanken und XML• NoSQL-Datenbanken• Graph-Datenbanken• Datenbanken und Datenschutz• Ontologien• Data Warehouses und Data Marts• Föderierte DB• objektrelationale DB• objektorientierte DB• ODBC-Schnittstelle• unformatierte DB (Dokumente) und automatische Indexierung• kommerzielle Datenbankanbieter• spezielle betriebswirtschaftliche Datenbankanwendungen

- Datenbanken in Medizin, Genetik, Biologie
- Datenbanken in Mathematik, Physik, Chemie
- Datenbanken in den Geisteswissenschaften, Digital Humanities
- Datenbanken in den Rechtswissenschaften
- SQL-Dialekte und SQL-Erweiterungen von DBMS-Produkten
- Open Source DB
- Recoverymaßnahmen
- Data-Dictionary-Systeme
- verteilte DB und Client-Server-Architekturen

Literatur

- Date, Christopher: Introduction to database systems. Reading, MA: Addison Wesley, 2003
- Kemper, A.; Eickler, A.: Datenbanksysteme. München: Oldenbourg, 2001
- Date, C.J.; Darwen, H.: A guide to the SQL standard. Reading, MA: Addison Wesley, 1997
- Pernul, Günter; Unland, Rainer: Datenbanken im Unternehmen. München: Oldenbourg, 2003

Leistungsnachweis

Referat (75 min, 50 %) und schriftliche Ausarbeitung (15 Seiten, 50 %) je Zweiergruppe.

Digitalisierung und Nachhaltigkeit

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer wählbar außerhalb Schwerpunkt
Modul	Digitalisierung und Nachhaltigkeit
Modulverantwortliche	Dina Barbian
Dozent(en)	Dina Barbian
Vorkenntnisse	Grundlagen der Wirtschaftswissenschaften, Supply Chain Management
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenz, 85 Stunden Vor- und Nachbereiten des Lehrstoffs, insb. Vorbereitung des Referats und Erstellen der Studienarbeit
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Seminaristischer Unterrichtsstil mit vielen Fallstudien
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	<ul style="list-style-type: none">• Identifikation der Möglichkeiten und Grenzen einer zunehmenden Digitalisierung für eine nachhaltige Entwicklung• Erlernen von informationstechnischen Lösungen zur Steigerung der Ressourcen- und Energieeffizienz• Kennenlernen von innovativen Anwendungssystemen zur Unterstützung des Menschen bei der Verrichtung von Arbeit
Lernziel	Die Studierenden können: <ul style="list-style-type: none">• die Begriffe "Digitalisierung" und "Nachhaltigkeit" erklären,• eine zunehmende Digitalisierung im Kontext von Nachhaltigkeit kritisch einordnen,• zu den gesellschaftlichen Herausforderungen informationstechnische Lösungen benennen,• wichtige Anwendungsfelder für eine Digitalisierung und zur Erreichung von Nachhaltigkeit identifizieren, und• die Risiken durch eine zunehmende Digitalisierung erläutern.
Schlüsselqualifikation	<ul style="list-style-type: none">• 40% Fachkompetenz• 30% Methodenkompetenz• 30% Sozialkompetenz
Lehrinhalte	<ol style="list-style-type: none">1. Grundlegende Begrifflichkeiten, Abgrenzung und Historie<ol style="list-style-type: none">1.1 Begriffe von Digitalisierung1.2 Entwicklung von Industrie 1.0 zu Industrie 4.01.3 Was ist Nachhaltigkeit?1.4 Zusammenhänge Ökonomie – Ökologie – Gesellschaft – Technologie1.5 Historie zu Nachhaltigkeit1.6 UN Sustainable Development Goals (SDGs) / UN-Entwicklungsziele1.7 Digitalisierung und Nachhaltigkeit: nachhaltig-digitale Organisationen

und Forderungen

2. Gesellschaftliche Herausforderungen
 - 2.1 Zunehmende Alterung in der Bevölkerung
 - 2.2 Fachkräftemangel
 - 2.3 Zunehmende Umweltverschmutzung, Klimawandel
 - 2.4 Schutz der kritischen Infrastrukturen
 - 2.5 Zunehmende Vernetzung: vernetzte Konsumgüter und Maschinen
3. Lösungsansätze
 - 3.1 Zunehmende Automatisierung als Lösung des Altersproblems?
 - 3.2 Einsatz von Robotik (z. B. als Hilfe in Altenheimen oder Hotels)
 - 3.3 Power-to-X-Technologien: Power-to-Gas (s. Audi e-Gas-Projekt)
 - 3.4 Leichtbauweise durch 3D-Druck zur Einsparung von Ressourcen
 - 3.5 Serviceroboter in Kaufhäusern und Banken; Sprachassistenten und Gesichtserkennung
 - 3.6 Abfallmanagementsysteme
 - 3.7 Modularer Aufbau elektronischer Produkte (siehe Fair- und ShiftPhone)
 - 3.8 Substitution nicht-erneuerbarer durch erneuerbare Ressourcen
 - 3.9 Sharing Economy: Car-sharing etc.
4. Ausgewählte Anwendungsfelder
 - 4.1 Zunehmender Einsatz von Drohnen, Sensorik, Robotik
 - 4.2 Smart Grid in der Energiewirtschaft: erneuerbare Energien
 - 4.3 Logistik und Transportsektor: eTrucks, Echtzeit-Nachverfolgung
 - 4.4 Industrielle Produktion: Predictive Maintenance, Smart Factory
5. Nutzeffekte
 - 5.1 Erhöhung der Ressourcen- und Energieeffizienz
 - 5.2 Assistenzen bei extremen Arbeitsbedingungen (Monotonie, Gefahren)
 - 5.3 Einsparung von Arbeitskräften, Rohstoffen und Zeit
6. Risiken durch eine zunehmende Digitalisierung
 - 6.1 Cyber-Kriminalität und Datensicherheit
 - 6.2 Schwierigkeiten bei der Normung und Rechtsrahmen
 - 6.3 Haftung für Fehler und Unfälle
 - 6.4 Zunehmende Arbeitslosigkeit unter Gering-Qualifizierten?

Literatur

Barbian, D., Ökonomie und Sustainable Development – Entwicklung eines Ansatzes zur Umsetzung von Nachhaltigkeit, Aachen 2001.

Mertens, P., Barbian, D. und Baier, S., Digitalisierung und Industrie 4.0 – Eine Relativierung, Berlin-Heidelberg 2017 (im Druck).

Barbian, D., Industrie 4.0 in der Lagerlogistik – Überblick, aktuelle Trends und Folgen für eine nachhaltige Entwicklung, in: Glock, C. und Grosse, E. (Hrsg.), Warehousing 4.0 – Technische Lösungen und Managementkonzepte für die Lagerlogistik der Zukunft, Lauda-Königshofen 2017, S. 17-36.

Barbian, D., Our common WASTE – solutions for a sustainable society, in: Plöhn, J. und Chobanov, G. (eds.), Sustainability and Welfare Policy in European Market Economies, Frankfurt am Main 2017, S. 127-145.

Barbian, D., Umweltmanagement – wozu?, Technik in Bayern 06/2016, S. 10-11.

Barbian, D., Cyber-Physical Systems - Can They Contribute to More Sustainability? in: Herzog, M. (ed.), Economics of communication: ICT driven fairness and sustainability for local and global marketplaces, Berlin 2015, S. 29-44.

Mertens, P. und Barbian, D., Beherrschung systemischer Risiken in weltweiten Netzen,

Informatik Spektrum 38 (2015) 4, S. 283-289.

Mertens, P. und Barbian, D., Die Wirtschaftsinformatik der Zukunft – auch eine Wissenschaft der Netze?, HMD – Praxis der Wirtschaftsinformatik, Heft 300 „Paradigmenwechsel“, Band 51, Heft 6, Dezember 2014, S. 729-743.

Leistungsnachweis

Die Seminarleistung besteht aus einem Referat (20 min., Gewichtung 50%) und einer Studienarbeit (20-30 Seiten, Gewichtung 50%).

Einführung in Business Intelligence

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer wählbar außerhalb Schwerpunkt
Modul	Einführung in Business Intelligence
Modulverantwortliche	Peter Rausch
Dozent(en)	Michael Lang
Vorkenntnisse	Grundlegende SAP-Kenntnisse können hilfreich sein, sind aber nicht zwingend notwendig
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenz, 85 Stunden Vor- und Nachbereiten des Lehrstoffs sowie Übungsaufgaben
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Vorlesung mit praktischer Übung
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Anwendungsbezogene und praxisorientierte Vorbereitung der Studierenden für Tätigkeiten im Bereich der Wirtschaftsinformatik. Aufzeigen der Querverbindungen zwischen Wirtschaft und IT.
Lernziel	Am Ende der Lehrveranstaltung sollen die Studierenden über fundierte Kenntnisse zu den in den Lehrinhalten angegebenen Themenfeldern verfügen. Die Studierenden sollen relevante Begriffe des BI kennen und die Fähigkeit haben, sicher mit ihnen umgehen können. Ferner sollen die Studierenden Konzepte des BI verstehen und das Erlernte in der Arbeitswelt anwenden können. Des Weiteren wird ein Einblick in die Technologien von BI-Systemen gegeben.
Schlüsselqualifikation	Befähigung zur selbständigen Anwendung von BI-Lösungen.
Lehrinhalte	Die Studierenden erhalten in dem Kurs einen vertiefenden Einblick zu folgenden Themenfeldern: <ul style="list-style-type: none">• Aufzeigen von Nutzungsmöglichkeiten in Unternehmen• Vermittlung von Software-Anwenderwissen in den Bereichen SAP BI und Pentaho• Analysemöglichkeiten mit Business Intelligence
Literatur	<ul style="list-style-type: none">• Rausch, Peter/ Sheta, Alaa /Ayesh, Aladdin (Hrsg.): Business Intelligence and Performance Management: Theory, Systems, and Industrial Applications, Springer Verlag, 2013• Kemper, H.G., Baars, H., Mehanna, W.: Business Intelligence – Grundlagen und praktische Anwendungen, akt. Aufl., Vieweg+Teubner Verlag, Wiesbaden• Chamoni, P./ Gluchowski, P. (Hrsg.): Analytische Informationssysteme: Business Intelligence-Technologien und –Anwendungen, akt. Aufl., Springer-Verlag• Bauer A., Günzel H. (Hrsg.): Data-Warehouse-Systeme, akt. Auflage, dpunkt, Heidelberg

Einsatz von Simulations-Software

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer wählbar außerhalb Schwerpunkt
Modul	Einsatz von Simulations-Software
Modulverantwortliche	Wolfgang Bremer
Vorkenntnisse	Grundlagen der Betriebswirtschaftslehre, insbesondere Interesse an Supply Chain Management, Grundkenntnisse in Statistik
Arbeitsaufwand	150 Stunden, davon 60 Stunden Präsenz sowie 90 Stunden Vor- und Nachbereiten des Lehrstoffes sowie Übungsaufgaben. Die Übungsaufgaben sind im Labor teilweise mit der Software zu lösen.
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Vorlesung + Übung im Labor (PC, Fallstudien)
Semesterturnus	Winter- und Sommersemester
Unterrichtssprache	Kurs in Deutsch und/oder Englisch (bitte nachfragen)
Beitrag zu den Zielen des Studiums	Stärkung des Abstraktionsvermögens. Ausbilden analytischer Fähigkeiten.
Lernziel	Kenntnis erlangen in den Instrumenten: Simulation und Modellierung. Befähigt sein, Supply-Chain und Serviceprozesse im ComputermodeLL abzubilden, deren Verhalten quantitativ im Vorfeld zu ermitteln und die Ergebnisse zu interpretieren.
Schlüsselqualifikation	Fähigkeit zu selbständigem Arbeiten. Abstraktion und Modellbildung. Analyse und Modellierung von Systemen. Kreatives Problemlösen
Lehrinhalte	1. Grundlagen betrieblicher Simulation 2. Einführung in die Software Arena 3. Aufbau einfacher Simulationsmodelle 4. Simulationslauf und Interpretation der Ergebnisse
Literatur	Altiok Tayfur et al.: Simulation Modeling and Analysis with Arena, Academic Press, 2007 Banks, Jerry et al.: Discrete Event Simulation, 5th edition, Pearson, 2010 Kelton, W. David: Simulation with Arena (5th Edition), Mc Graw Hill Robinson, Stewart: Simulation, 2nd edition, Palgrave Macmillian, 2014
Leistungsnachweis	Schriftliche Prüfung (90 min)

Electronic Commerce

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer wählbar außerhalb Schwerpunkt
Modul	Electronic Commerce
Modulverantwortliche	Patricia Brockmann
Dozent(en)	Patricia Brockmann
Vorkenntnisse	Inhalte der (Teil-)Module Grundlagen der Wirtschaftsinformatik
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenz 85 Stunden Vor- und Nachbereiten des Lehrstoffs und Prüfungsvorbereitung
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Seminar
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs nur in Englisch
Beitrag zu den Zielen des Studiums	Integration von Konzepten des Geschäftsprozessmanagements mit technischen Fähigkeiten, um IT-unterstützt betriebliche Informationssysteme zu analysieren und entwickeln.
Lernziel	Die Studierenden erhalten in dem Kurs einen vertiefenden Einblick und untersuchen sinnvoller Anwendungen des Electronic Commerce in Unternehmen
Schlüsselqualifikation	Analytische Fähigkeiten im Bereich Anforderungsspezifikationen Entwicklung von technischen Konzepten um Problemlösungen zu generieren Präsentationsfähigkeiten Praktische Anwendung englischer Sprache in einer technischen Umgebung Teamfähigkeiten
Lehrinhalte	Projektbezogene Anwendung Konzepte der digitalen Transformation zur Entwicklung eines eigenen e-Commerce Geschäftsplans <ul style="list-style-type: none">• Design Thinking bei der Entwicklung neuer Geschäftsmodelle• M-Commerce: mobile und ortsbezogene Anwendungen• Digital Marketing• Search Engine Optimization• Anwendung vom Machine Learning in E-Commerce• Gamification in E-Commerce• Evaluierung Benutzerschnittstellen mit Eye-Tracking• Internationale Aspekte des E-Commerce• Ethische Aspekte in E-Commerce Weitere Themen können von Studierenden vorgeschlagen werden.
Literatur	"E-Commerce: Business, Technology, Society", Laudon, K., Traver, G. Prentice Hall

2019.

Leistungsnachweis

Seminarleistung: Erstellung, Präsentation und schriftliche Dokumentation eines Konzepts für eine E-Commerce-Anwendung, auf Englisch oder auf Deutsch (10 Seiten) mit Referat (30 Min). Gewichtung 60% Hausarbeit 40% Referat.

Finanzmathematik

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer wählbar außerhalb Schwerpunkt
Modul	Finanzmathematik
Modulverantwortliche	Yvonne Stry
Dozent(en)	Yvonne Stry
Vorkenntnisse	Mathematik I und II
Arbeitsaufwand	75 Stunden, davon: 30 Präsenz 45 Stunden Vor- und Nachbereitung
Leistungspunkte	2.5
Semesterwochenstunden	2
Veranstaltungstyp	Seminaristischer Unterricht mit zahlreichen Beispielen und Übungen
Semesterturnus	Winter- bzw. Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Finanzmathematik als Anwendung der Mathematik in der Wirtschaft
Lernziel	Einführung in die Methoden der Finanzmathematik
Schlüsselqualifikation	Anwendung einfacher mathematischer Formeln bei Geldgeschäften
Lehrinhalte	<ul style="list-style-type: none">• Arithmetische und geometrische Folgen und Reihen• Zinsrechnung (einfache Verzinsung, Zinseszins, gemischte Zinsrechnung, unterjährliche Verzinsung, Effektivzins)• Rentenrechnung (vorschüssige und nachschüssige Rente, Rentenbarwerte, Rentenendwerte)• Tilgungsrechnung (Ratentilgung, Annuitätentilgung)• Ausblick
Literatur	Schwenkert/Stry, Finanzmathematik kompakt, Springer, 2. Auflage 2016
Leistungsnachweis	Klausur 90 min

IT- und Wirtschaftsrecht

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer wählbar außerhalb Schwerpunkt
Modul	IT- und Wirtschaftsrecht
Modulverantwortliche	Alexander Kübler-Kreß
Vorkenntnisse	Schulwissen in Wirtschaft und Recht
Leistungspunkte	2.5
Semesterwochenstunden	2
Semesterturnus	Winter- und Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Befähigung zur logischen Analyse von Problemen aus rechtlicher Sicht als Teil der für Informatiker maßgebenden Projektbetrachtung.
Lernziel	Kenntnis wichtiger rechtlichen Rahmenbedingungen und die Fähigkeit, für die Informationsverarbeitung typische Vorgänge rechtlich bewerten und gestalten zu können.
Schlüsselqualifikation	Fähigkeit zu komplexem Denken und zur exakten logischen Analyse
Lehrinhalte	Ausgewählte Themen aus Bereichen des Rechtes mit besonderer Relevanz für Entwicklung, Einführung und Betrieb von IT-Systemen. Mögliche Themengebiete sind z.B. das Wirtschaftsprivatrecht, IT-Recht, Datenschutzrecht, Medienrecht, Markenrecht.
Literatur	Themenabhängig, Bekanntgabe im Kurs
Leistungsnachweis	Klausur (90 min)

Spieltheorie

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer wählbar außerhalb Schwerpunkt
Modul	Spieltheorie
Modulverantwortliche	Ralf-Ulrich Kern
Dozent(en)	Ralf-Ulrich Kern
Vorkenntnisse	Mathematik-Vorlesungen; Kenntnisse in OR vorteilhaft.
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenzzeit, 85 Stunden für Vor- und Nachbereitung des Lehrstoffs, Vorbereitung auf die Prüfung.
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Vorlesung
Semesterturnus	Wintersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Mathematische Grundlagen für weiterführende Veranstaltungen über Methoden zur Entscheidungsunterstützung.
Lernziel	Kenntnis wichtiger Begriffe der mathematischen Spieltheorie: Einführung in strategisches Denken; Einführung in die Theorie der Verhandlungslösungen. Fähigkeit, Verhandlungspositionen einzuschätzen.
Schlüsselqualifikation	Grundlagen des strategischen Denkens und von Verhandlungsstrategien.
Lehrinhalte	<p>Die mathematische Spieltheorie befasst sich mit optimalen Entscheidungen bei Vorhandensein mehrerer Entscheidungsträger, anders ausgedrückt, mit optimalem strategischem Handeln. Die Bezeichnung "Spieltheorie" kommt daher, dass die ersten Studien anhand von Gesellschaftsspielen betrieben wurden; inzwischen findet die Spieltheorie Anwendung in Ökonomie, Politik, Bio- und Sozialwissenschaften.</p> <p>In Teil I der Vorlesung (kompetitive Spieltheorie) soll es um Verhalten bei Konkurrenz gehen, wenn also die Interessen der Entscheidungsträger (Spieler) entgegengesetzt gerichtet sind. Man versucht, Handlungsalternativen (Strategien) zu bestimmen, die unter Berücksichtigung des Verhaltens der Mitspieler den eigenen Gewinn maximieren.</p> <p>Stichwörter zum Inhalt: Spiele in extensiver Form, Spiele in Normalform, Matrixspiele mit (im)perfekter Information, mit/ohne Zufallseinfluss, Nullsummenspiele. Gleichgewichtspunkte, gemischte Strategien. Satz von Nash, Lösung von Matrixspielen.</p> <p>In Teil II, der kooperativen Spieltheorie, werden Gewinne betrachtet, die im Gegensatz zur Konkurrenztheorie durch Verfolgung gleichgerichteter Interessen erzielt werden. Das Augenmerk liegt daher nicht auf der Wahl der richtigen Strategie, um einen</p>

bestimmten Gewinn zu realisieren, sondern auf der Aufteilung des gemeinsam erzielten Gewinns unter den (typischerweise mehr als zwei) kooperierenden Spielern. Dabei spielen Gesichtspunkte wie Fairness, Gerechtigkeit, Durchsetzbarkeit, Stabilität eine Rolle.

Man betrachtet Spiele nicht mehr in der Normalform, sondern in der "charakteristischen Form" und versucht, daraus Verhandlungslösungen, also Vorschläge zur Gewinnaufteilung, herzuleiten, entweder in Form von eindeutigen Funktionen oder von mengenwertigen Abbildungen.

Stichwörter zum Inhalt:

Verhandlungslösungen (z. B. Core, Shapley-Wert, Nash-Wert).
Individuelle Rationalität, Verhandlungsstärke, Koalitionsbildung.

Dazu zahlreiche (auch unterhaltsame) Beispiele.

Literatur

- Güth, Werner: Spieltheorie und ökonomische (Bei)Spiele. Springer-Verlag Berlin usw. (1999). ISBN 3-540-54921-8.
- Holler, Manfred J., Illing, Gerhard: Einführung in die Spieltheorie. Springer-Verlag Berlin usw. (7. Aufl. 2009). ISBN 978-3-540-69372-7.

Leistungsnachweis

Klausur (90 min), leider kein Spielturnier :-)

Technologien und betriebswirtschaftliche Bedeutung von Industrie 4.0

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer wählbar außerhalb Schwerpunkt
Modul	Technologien und betriebswirtschaftliche Bedeutung von Industrie 4.0
Modulverantwortliche	Heidi Schuhbauer
Dozent(en)	Heidi Schuhbauer
Vorkenntnisse	Grundlagen der Betriebswirtschaftslehre Supply Chain Managment
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenz sowie 85 Stunden Vor- und Nachbereiten des Lehrstoffes sowie Übungsaufgaben
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Seminar
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Methoden zur Stärkung des Abstraktionsvermögens und zur Stärkung der analytischen Fähigkeiten
Lernziel	<ul style="list-style-type: none">• Fachkompetenz: Die Studierenden verstehen das Konzept von Industrie 4.0, können es entwicklungsgeschichtlich einordnen, kennen die einzelnen Technologien von Industrie 4.0 sowie deren Einsatzpotenziale und setzen sich mit den Konsequenzen für die Bildung auseinander• Methodenkompetenz: Fähigkeit eine betriebswirtschaftliche Problemstellung zu analysieren und dafür eine IT-Lösung zu konzipieren und diese kritisch zu evaluieren; Fähigkeit einen wissenschaftlichen Vortrag auszuarbeiten und zu halten sowie eine wissenschaftliche Ausarbeitung dazu anzufertigen.• Soziale Kompetenz: Organisation, Planung, Präsentationstechnik
Schlüsselqualifikation	Fähigkeit zu selbständigem Arbeiten; Strategien des Wissenserwerbs; Analyse und Klassifikation von Problemen; kreatives Problemlösen; Präsentationskompetenz
Lehrinhalte	<p>Die Studierenden erhalten in dem Kurs einen vertiefenden Einblick zu folgenden Themenfeldern:</p> <ul style="list-style-type: none">• Begriffe um Industrie 4.0 und Entwicklungsgeschichte• Technologien um Industrie 4.0 und Einsatzmöglichkeiten mit betriebswirtschaftlicher Betrachtung (Cyberphysische Systeme, Ubiquitous Computing, Internet der Dinge und Dienste, Schnittstellen für Maschine-zu-Maschine-Kommunikation und für die Mensch-Maschine-Interaktion, Cloud Computing, Big Data und Analytics Dienste)• Konsequenzen aus Industrie 4.0 für die Aus- und Weiterbildung
Literatur	Roth, A. (Hrsg.), Einführung und Umsetzung von Industrie 4.0 – Grundlagen, Vorgehensmodell und Use Cases aus der Praxis, Heidelberg 2016.

Leistungsnachweis

Seminarleistung (Referat 50 Minuten, Handout (4 Seiten), Studienarbeit mit dem Plenum 20 Minuten), Video Erstellung; Gewichtung Referat mit Handout 60%, Studienarbeit 30%, Video 10 %

Bachelorarbeit

Studiengang	Bachelor Informatik 2. Studienabschnitt Abschlussarbeit
Modul	Bachelorarbeit
Vorkenntnisse	Alle Fähigkeiten, die das Studium vermittelt
Arbeitsaufwand	2 Monate
Leistungspunkte	12
Semesterwochenstunden	0
Veranstaltungstyp	Selbständiges Arbeiten
Semesterturnus	Winter- bzw. Sommersemester
Unterrichtssprache	Kurs in Deutsch und/oder Englisch (bitte nachfragen)
Beitrag zu den Zielen des Studiums	Die weitgehend selbständige Arbeit auf wissenschaftlicher Grundlage ist eines der wesentlichen Ziele eines jeden Bachelor-Studiums.
Lernziel	Die Fähigkeit, ein praxisbezogenes Problem aus den Gebieten der Pflicht- und fachwissenschaftlichen Wahlpflichtmodule, auch fachübergreifend, selbständig und auf wissenschaftlichen Grundlagen methodisch zu bearbeiten.
Schlüsselqualifikation	Fähigkeit zu selbständigem Arbeiten auf wissenschaftlicher Grundlage, technisches Schreiben, Analyse und Klassifikation von Problemen, kreatives Problemlösen
Lehrinhalte	Anleitung zu selbständiger Arbeit auf wissenschaftlichen Grundlagen
Literatur	Themenbezogen
Leistungsnachweis	Zur Bachelorarbeit kann sich anmelden, wer den praktischen Teil des praktischen Studiensemesters mit Erfolg abgelegt und insgesamt 160 Leistungspunkte erreicht hat. Die Bachelorarbeit kann mit Genehmigung der Prüfungskommission auch in Englisch oder in einer anderen Fremdsprache verfasst werden. Leistungsnachweis ist die termingerecht abgegebene Arbeit.

Seminar zur Bachelorarbeit

Studiengang	Bachelor Informatik 2. Studienabschnitt Abschlussarbeit
Modul	Seminar zur Bachelorarbeit
Vorkenntnisse	Praktisches Studiensemester
Arbeitsaufwand	60 Stunden, davon: 30 Stunden Präsenzzeit
Leistungspunkte	2
Semesterwochenstunden	2
Veranstaltungstyp	2 SWS Seminar
Semesterturnus	Winter- bzw. Sommersemester
Unterrichtssprache	Kurs in Deutsch und/oder Englisch (bitte nachfragen)
Beitrag zu den Zielen des Studiums	Die Lehrveranstaltung fördert die Kompetenz, die Ergebnisse der eigenen Arbeit zu präsentieren und in Diskussionen zu erläutern. Zuhörer erhalten einen umfassenden Überblick über Arbeitsgebiete an der Hochschule und berufliche Perspektiven in Unternehmen.
Lernziel	Fähigkeit, eine eigene fachlich-wissenschaftliche Arbeit zu präsentieren und zu verteidigen. Fähigkeit, spezielle fachliche Inhalte zu verstehen und zu hinterfragen.
Schlüsselqualifikation	Fähigkeit zu selbständigem Arbeiten, Präsentationskompetenz, Zeitmanagement, praktische Erfahrung
Lehrinhalte	Präsentation, Diskussion und Disputation. Fachliche Inhalte entsprechend den vorgestellten Themen der Bachelor-Arbeiten.
Leistungsnachweis	Jeder Bachelor-Kandidat hält im Seminar einen Vortrag von ca. 45 min Dauer (einschließlich Diskussion) zum Thema seiner Bachelorarbeit. Er muss vor Anmeldung seiner Bachelorarbeit an zwei je 90-minütigen Kursen zu den Themen Wissenschaftliches Schreiben und Literaturrecherche teilnehmen und bis zum Abschluss seiner Bachelorarbeit mindestens 8 Vorträge im Seminar besuchen. Der Studierende führt den Nachweis hierzu selbst und muss diesen vor der Anmeldung bzw. vor der Bewertung seiner Bachelorarbeit dem Erstprüfer seiner Bachelorarbeit vorlegen. Einzelheiten zum organisatorischen Ablauf werden auf www.in.th-nuernberg.de unter -> Studierende -> Infos zum Studium -> Abschlussarbeiten bekanntgegeben. Das Seminar zur Bachelorarbeit wird benotet, die Note wird im Zeugnis ausgewiesen und geht in die Gesamtnote ein.

Hilfsmittel

- 1) keine Hilfsmittel
- 2) unbeschränkt
- 3) Vorlesungsmitschrift
- 4) Vorlesungsumdrucke
- 5) selbstgefertigte Arbeitsunterlagen (DinA4 Seitenzahl)
- 6) mathematische Formelsammlung
- 7) Taschenrechner, nicht programmierbar (Taschenrechner mit vollständiger alphanumerischer Tastatur und/oder Graphikdisplay sind nicht erlaubt)
- 8) Gesetzestexte, z.B. BGB, UrhG, PatG, UWG, Betr.VG, BDSG, StGB, Stopp, TKG
- 9) Lehrbuch