

# Modulhandbuch für den Studiengang Bachelor Informatik

BIN/BWIN Studienordnung ab WS 2021/2022

<b>Begriffserläuterungen</b> .....	3
<b>1. Studienabschnitt</b> .....	4
<b>allgemeines Modul</b> .....	4
Allgemeinwissenschaftliche Wahlpflichtmodule .....	5
<b>Theoretische Grundlagen</b> .....	5
Algorithmen und Datenstrukturen .....	6
Grundlagen der Informatik .....	8
Mathematik 1: Algebra .....	10
Mathematik 2: Analysis .....	11
Theoretische Informatik .....	12
<b>Praktische Informatik</b> .....	13
Objektorientierte Programmierung .....	14
Prozedurale Programmierung .....	15
<b>Systeminfrastrukturen</b> .....	16
Computerarchitektur .....	17
<b>Allgemeinwissenschaften</b> .....	18
Conversational English .....	19
<b>2. Studienabschnitt</b> .....	19
<b>Theoretische Grundlagen</b> .....	19
Mathematik 3: Statistik .....	20
<b>Softwaremethoden und -technologien</b> .....	20
Datenbanken .....	21
IT-Anwendungen (Bachelor IN) .....	23
Programmiersprachen .....	24
Software Engineering .....	25
Software Engineering Praktikum .....	26
Softwarearchitektur .....	27
<b>Systeminfrastrukturen</b> .....	28
Betriebssysteme .....	29
Rechnerkommunikation und -netze .....	30
<b>Vertiefende Wahlpflichtfächer</b> .....	30
Grundlagen der Wirtschaftswissenschaften .....	31
Informationssicherheit und ihre Grundlagen .....	32
Kryptographie und Informationssicherheit .....	34
Machine Learning .....	36
Mathematische Methoden für maschinelles Lernen .....	37
Mensch-Computer-Interaktion .....	39
Natural Language Processing .....	40
Webanwendungen .....	42
<b>Praktisches Studiensemester</b> .....	43
Praktikum .....	44
Praxisseminar .....	45
Projektmanagement .....	46
<b>Fachwissenschaftliche Wahlpflichtfächer</b> .....	47
<i>Achtung: Das zukünftig regelmäßige bzw. das tatsächliche Angebot der folgenden Fächer kann variieren !</i> .....	47
<b>Schwerpunkt-FWPF (min. 16SWS/20LP, max. SWS/LP)</b> .....	47
Aktuelle Entwicklungen des Computer Designs .....	48
Algorithmische Graphentheorie .....	50
Angewandtes Projektmanagement .....	51
Autonome mobile Roboter .....	52
Betriebssysteme für einfache Mikrocontroller .....	54
Clean Code in Java-Projekten .....	56
Computational Intelligence .....	58
Datenfusion .....	60
Datenmodellierung mit XML .....	62
Datenschutz .....	64

Version 6, 05.04.2024 (IN)

IN\_1784\_V6\_VO Modulhandbuch B IN ab 21

Einführung in die Funktionale Programmierung .....	66
Embedded Systems .....	67
Graphical User Interface Design and Information Visualization .....	69
Grundlagen Content-Management-Systeme .....	70
Grundlagen der Computerspielentwicklung .....	72
Grundlagen der Software Security .....	74
Grundlagen des maschinellen Lernens .....	76
Grundlagen des Softwaretests .....	77
Grundlagen des Softwaretests (BLOCK) .....	79
In-Memory Computing am Beispiel von SAP HANA .....	81
Knowledge Graphen .....	83
Kryptographische Algorithmen und Systeme .....	85
Lean Startup .....	87
Moderne Versionen der Logik .....	89
Netzwerksicherheit .....	91
Programmieren von Mikrocontrollern .....	94
Programmierung und Technologie betrieblicher Standardsoftware (SAP-Praktikum) .....	96
Requirements Engineering .....	97
Simulation von Algorithmen des Maschinellen Lernens für autonomes Fahren .....	99
Software-Ergonomie in der Medizintechnik .....	101
Technical Computing .....	102
Testverfahren für komplexe Softwaresysteme .....	103
Text Analytics .....	105
Verteilte Systeme - Grundlagen und Algorithmen .....	107
Visualisierung .....	109
Web Application Security .....	111
<b>wählbar außerhalb Schwerpunkt</b> .....	<b>112</b>
Analyse von Data Mining Use Cases aus dem Bereich Marketing und Finanzierung .....	113
Digital Government .....	115
Digitale Bauwirtschaft .....	116
Digitalisierung und Nachhaltigkeit .....	118
Finanzmathematik .....	121
Informatikethik .....	122
Informationssicherheitsmanagement .....	125
Introduction to Bionic Computation in Business .....	126
Optimierung 1 .....	128
Spieltheorie .....	130
<b>Abschlussarbeit</b> .....	<b>131</b>
Bachelorarbeit .....	132
Seminar zur Bachelorarbeit .....	133
<b>Hilfsmittel</b> .....	<b>134</b>

# 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"><li>• Präsenzzeit,</li><li>• Zeit für die Vor- und Nachbereitung des Vorlesungsstoffs,</li><li>• Zeit für die Vorbereitung von Vorträgen und Präsentationen,</li><li>• Zeit für die Erstellung eines Projekts,</li><li>• Zeit für die Ausarbeitung einer Studienarbeit,</li><li>• Zeit für notwendiges Selbststudium,</li><li>• Zeit für die Vorbereitung auf mündliche oder schriftliche Prüfungen.</li></ul> <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.

## Allgemeinwissenschaftliche Wahlpflichtmodule

Studiengang	Bachelor Informatik 1. Studienabschnitt allgemeines Modul
Modul	Allgemeinwissenschaftliche Wahlpflichtmodule
Modulverantwortliche	All Professors of Faculty AMP perm. for lecture / Alle für Fach zugew. AMP-Professoren
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 1. 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 Übungsaufgaben und Prüfungsvorbereitung
Leistungspunkte	7
Semesterwochenstunden	6
Veranstaltungstyp	4 SWS Vorlesung, 2 SWS Übung
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"><li>• T.H.Cormen, C.E.Leiserson, R.L.Rivest, C.Stein: Algorithmen – Eine Einführung, 4. Aufl., 2013, De Gruyter Oldenbourg</li><li>• T. Ottmann, P. Widmayer: Algorithmen und Datenstrukturen, 6. Aufl., Springer Vieweg, 2017.</li><li>• V. Heun: Grundlegende Algorithmen: Einführung in den Entwurf und die Analyse effizienter Algorithmen, 2.Aufl., Braunschweig, Wiesbaden, 2003, Vieweg.</li><li>• D.E. Knuth: The Art of Computer Programming, Volume 1+3: Fundamental</li></ul>

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, 2.Aufl., Springer, 2021.

Leistungsnachweis

Schriftliche Prüfung (90 min).

Zulassungsvoraussetzung

# Grundlagen der Informatik

Studiengang	Bachelor Informatik 1. Studienabschnitt Theoretische Grundlagen
Modul	Grundlagen der Informatik
Modulverantwortliche	Rainer Weber, Thomas Fuhr, Jörg Roth, Michael Zapf, Florian Gallwitz
Dozent(en)	Cristian Axenie, Rainer Weber
Vorkenntnisse	Schulwissen
Arbeitsaufwand	210 Stunden, davon: 90 Stunden Kontaktzeit 95 Stunden Vor- und Nachbereitung des Lehrstoffs 25 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"><li>• Repräsentation positiver ganzer Zahlen in verschiedenen Stellenwertsystemen</li><li>• Negative ganze Zahlen in verschiedenen Stellenwertsystemen</li><li>• Reelle Zahlen im Standardformat (IEEE-754)</li><li>• Rechenoperationen und Konvertierungen</li><li>• Boolesche Algebren</li><li>• Boolesche Ausdrücke und Funktionen</li><li>• Schaltnetze und Schaltwerke</li><li>• Allgemeine Computerarchitektur</li><li>• Maschinensprache am Beispiel der MIPS-Architektur</li><li>• Assemblersprache der MIPS-Architektur</li></ul>
Literatur	<ul style="list-style-type: none"><li>• R. Klar: Digitale Rechenautomaten, Berlin, 1988, Walter de Gruyter.</li></ul>



- 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 Hochschulschriften für Technik und Wirtschaft. Nürnberg, 2001, Nano-Verlag.

Leistungsnachweis

Schriftliche Prüfung (90 min)

Zulassungsvoraussetzung

# Mathematik 1: Algebra

Studiengang	Bachelor Informatik 1. Studienabschnitt Theoretische Grundlagen
Modul	Mathematik 1: Algebra
Modulverantwortliche	Herbert Urban, Alexander Hufnagel, Elke Wilczok, Christian Scherr
Dozent(en)	Matthias Börger, Jochen Gorski, Christian Scherr, Herbert Urban, Edgar Wermuth
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 2: Analysis

Studiengang	Bachelor Informatik 1. Studienabschnitt Theoretische Grundlagen
Modul	Mathematik 2: Analysis
Modulverantwortliche	Herbert Urban, Alexander Hufnagel, Elke Wilczok, Christian Scherr
Dozent(en)	Jürgen Bolik, Jochen Gorski, Klaus Greipel, Alexander Hufnagel, Christian Scherr, Edgar Wermuth
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"><li>• G. Teschl, S. Teschl: Mathematik für Informatiker (Springer)</li><li>• K. Meyberg, P. Vachenauer: Höhere Mathematik (Springer)</li></ul>
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
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"><li>• Dirk. W. Hoffmann: Theoretische Informatik, 3. Auflage, 2015, Carl Hanser Verlag.</li><li>• G. Vossen, K.-U. Witt: Grundkurs theoretische Informatik: Eine anwendungsbezogene Einführung, 6. Auflage. Braunschweig, 2016, Springer Vieweg.</li><li>• U. Hedstück: Einführung in die theoretische Informatik: Formale Sprachen und Automatentheorie, 5. Auflage. 2012, Oldenbourg Wissenschaftsverlag.</li><li>• J.E.Hopcroft &amp; R. Motwani &amp; J.D.Ullman: Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie; 3. Auflage, 2011, Pearson Studium.</li><li>• H. Knebl: Informationsverarbeitende Systeme. Nürnberger Hochschulschriften für Technik und Wirtschaft. Nürnberg, 2003, Nano-Verlag.</li></ul>



## Objektorientierte Programmierung

Studiengang	Bachelor Informatik 1. Studienabschnitt Praktische Informatik
Modul	Objektorientierte Programmierung
Modulverantwortliche	Uwe Wienkop, Alexander Schneider, Timo Götzelmann, Bartosz von Rymon Lipinski, Birgit Kraft
Dozent(en)	Yilmaz Duman, Birgit Kraft, Matthias Meitner, Alexander Schneider, Ingo Scholz, Simon Seibt, Uwe Wienkop
Vorkenntnisse	Prozedurale Programmierung
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 Studienprogramme der Studiengänge Informatik, Medieninformatik und Wirtschaftsinformatik haben mit der Anwendungsentwicklung von Software zu tun. Das Modul OOP führt die in PP gelegten Grundlagen um i.w. objektorientierte Aspekte fort.
Lernziel	Vertiefung und Erweiterung der Fähigkeiten, die in PP erworben wurden.
Schlüsselqualifikation	Fähigkeit zu komplexem Denken Ausdauer bei Problemlösungen
Lehrinhalte	Fortsetzung und Erweiterung der Lehrinhalte von Prozedurale Programmierung: dynamische Datenstrukturen, insbesondere verkettete Listen, Operatoren, Nutzung von Klassenbibliotheken, Ausnahmen und ihre Behandlung.
Literatur	<ul style="list-style-type: none"><li>• Kühnel, Andreas, <i>Visual C# 2012</i>, Rheinwerk Computing</li><li>• Doberenz, Walter, <i>Visual C# 2015</i>, Hanser Verlag</li><li>• Hanspeter Mössenböck: <i>Kompaktkurs C# 6.0</i>, dpunkt Verlag</li><li>• Skript.</li></ul>
Leistungsnachweis	Schriftliche Prüfung (90 min)
Zulassungsvoraussetzung	

# Prozedurale Programmierung

Studiengang	Bachelor Informatik 1. Studienabschnitt Praktische Informatik
Modul	Prozedurale Programmierung
Modulverantwortliche	Birgit Kraft, Jens Albrecht, Friedhelm Stappert
Dozent(en)	Jens Albrecht, Birgit Kraft, Ingo Scholz, Friedhelm Stappert, Thomas Ulrich
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 Prozedurale Programmierung schafft dafür Grundlagen.
Lernziel	Nach dem erfolgreich Abschluss des Moduls Prozedurale Programmierung sind die Studierenden in der Lage <ul style="list-style-type: none"><li>• grundlegende Kontroll- und Datenstrukturen einer Programmiersprache sicher einzusetzen</li><li>• einfache algorithmische Probleme zu analysieren und Lösungen dafür zu implementieren</li><li>• Elemente von Programmiersprachen wie Syntax, Namensbindung, Typsystem, Speicherstrukturen, Funktionsaufrufe und Parameterübergabe zu erkennen und zu erklären</li></ul>
Schlüsselqualifikation	<ul style="list-style-type: none"><li>• Fähigkeit zu komplexem Denken</li><li>• Ausdauer bei Problemlösungen</li></ul>
Lehrinhalte	<ul style="list-style-type: none"><li>• Begriffe zur Datenverarbeitung, Syntax und Semantik der Sprachelemente: Ablaufstrukturen, Datenstrukturen, Objekte, Module, Iteration und Rekursion.</li><li>• Entwicklungsmethoden: Entwicklungsumgebung, Entwicklung und Darstellung von Daten- und Ablaufstrukturen, strukturierter Entwurf und Implementierung, Dokumentation, Test.</li></ul>
Literatur	<ul style="list-style-type: none"><li>• Theis: Einstieg in C# mit Visual Studio 2019, Rheinwerk Computing, 2019</li><li>• Doberenz u.a.: <i>Visual C# 2017</i>, Grundlagen, Profiwissen und Rezepte, Hanser Verlag, 2017</li><li>• Mössenböck: <i>Kompaktkurs C# 6.0</i>, dpunkt Verlag, 2016</li></ul>

- Skript.

Leistungsnachweis

Schriftliche Prüfung (90 min)

Zulassungsvoraussetzung

Erfolgreiche Teilnahme am Praktikum.



# Computerarchitektur

Studiengang	Bachelor Informatik 1. Studienabschnitt Systeminfrastrukturen
Modul	Computerarchitektur
Modulverantwortliche	Axel Hein, Michael Zapf
Dozent(en)	Axel Hein, Michael Zapf
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"><li>• Darstellung wichtiger Rechnerstrukturen, einschließlich Prozessoren, Peripheriegeräten, Speicherorganisation, Magnetische Festplatten, Flash Speicher, Caching, Superskalarität, Pipelining, Multithreading, Cache-Kohärenz und Verbindungsstrukturen.</li><li>• Mikroarchitekturen und Instruktionssatz-Architekturen von Prozessoren.</li><li>• Bewertung der Leistungsfähigkeit und Fehlertoleranz von Rechnersystemen.</li><li>• Nutzung von Parallelität.</li></ul>
Literatur	<ul style="list-style-type: none"><li>• Andrew S. Tanenbaum: Structured Computer Organization (deutsch: Computerarchitektur), 2012, Prentice Hall, 6th edition</li><li>• William Stallings: Computer Organization and Architecture, New Jersey, 2012, Addison Wesley, 9th edition</li><li>• John L. Hennessy, David A. Patterson: Computer Organization and Design – The Hardware/Software Interface, 2013, Morgan Kaufmann Publishers Inc., 5th edition</li><li>• John L. Hennessy, David A. Patterson: Computer Architecture – A Quantitative Approach, San Francisco, 2012, Morgan Kaufmann Publishers Inc. / Elsevier, 5th edition</li></ul>

Leistungsnachweis

siehe Festlegung in SPO

Zulassungsvoraussetzung

# Conversational English

Studiengang	Bachelor Informatik 1. Studienabschnitt Allgemeinwissenschaften
Modul	Conversational English
Modulverantwortliche	Anita Vrzina
Vorkenntnisse	Schulwissen
Arbeitsaufwand	60 Stunden, davon: 30 Stunden Präsenz 30 Stunden Vor- und Nachbereitung des Lehrstoffs und Prüfungsvorbereitung
Leistungspunkte	2
Semesterwochenstunden	2
Veranstaltungstyp	2 SWS seminaristischer Unterricht
Semesterturnus	Winter- und Sommersemester
Unterrichtssprache	Kurs nur in Englisch
Beitrag zu den Zielen des Studiums	Fähigkeit zur Einarbeitung in fachliche Kontexte sowie entsprechende Konversation in englischer Sprache
Lernziel	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"><li>• Bruckenmaier, Stephan. 2008. <i>Grammatik kurz und bündig – Englisch</i>. Stuttgart: Klett Verlag (Pons).</li><li>• Clark, David. 1997. <i>Englisch Grammatik</i>. Pocket Teacher. Berlin: Cornelsen Verlag.</li><li>• Hoffmann, Hans &amp; Marion Hoffmann. 2006. <i>Kurzgrammatik Englisch. Zum Nachschlagen und Üben</i>. München: Max Hueber.</li><li>• Jacob, Rainer. 2008. <i>Kurzgrammatik. Kompakt-Wissen Englisch</i>. Freising: Stark Verlag.</li><li>• Walther, Lutz. 2008. <i>Kurzgrammatik Englisch</i>. Berlin: Langenscheidt.</li><li>• Weis, Erich. 1977. <i>Grund- und Aufbauwortschatz Englisch</i>. Stuttgart: Klett. (Nachdruck – 1986)</li></ul>
Leistungsnachweis	Teilnahme an mind. 80% der Termine (siehe APO)
Zulassungsvoraussetzung	

## Mathematik 3: Statistik

Studiengang	Bachelor Informatik 2. Studienabschnitt Theoretische Grundlagen
Modul	Mathematik 3: Statistik
Modulverantwortliche	Hans Delfs, Alexander Hufnagel
Dozent(en)	Alexander Hufnagel, 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, Jens Albrecht
Dozent(en)	Jens Albrecht, Patricia Brockmann, Birgit Kraft
Vorkenntnisse	Lehrveranstaltungen des ersten Studienjahres, insbesondere Grundlagen der Informatik, Programmieren.
Arbeitsaufwand	210 Stunden, davon: 95 Stunden Präsenzzeit, 115 Stunden für Vor- und Nachbereitung des Lehrstoffs, Bearbeitung der Übungs- und Praktikumsaufgaben und Prüfungsvorbereitung
Leistungspunkte	7
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	Kenntnis und Verständnis wichtiger Methoden und Techniken zur Modellierung und Realisierung von Datenbanken, Überblick über Architektur, Funktionsweise und Einsatz von Datenbanksystemen Fähigkeit zur professionellen Entwicklung von Datenbankanwendungen
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.
Literatur	<ul style="list-style-type: none"><li>• Date, Christopher: Introduction to database systems. Reading, MA: Addison Wesley, 2003</li><li>• Kemper, A.; Eickler, A.: Datenbanksysteme. München: Oldenbourg, 2001</li><li>• Date, C.J.; Darwen, H.: A guide to the SQL standard. Reading, MA: Addison Wesley, 1997</li><li>• Pernul, Günter; Unland, Rainer: Datenbanken im Unternehmen. München: Oldenbourg, 2003</li></ul>



## IT-Anwendungen (Bachelor IN)

Studiengang	Bachelor Informatik 2. Studienabschnitt Softwaremethoden und -technologien
Modul	IT-Anwendungen (Bachelor IN)
Modulverantwortliche	Peter Trommler
Dozent(en)	Florian Gallwitz, Korbinian Riedhammer, Jörg Roth, Friedhelm Stappert, 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	7
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)	Matthias Teßmann
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"><li>• Syntaxanalyse, semantische Analyse, Codegenerierung</li><li>• Werte, Typen, Namen, Bindungen, Speicherabbildung, Kontrollfluss, Typsysteme, prozedurale, Daten- und generische Abstraktion.</li><li>• Prozedurale, objektorientierte und funktionale Programmierung, Skript-Sprachen.</li><li>• Virtuelle Maschinen, Typsysteme zur Laufzeit, Komponenten, Speicherverwaltung, Code-Sicherheit.</li></ul>
Literatur	<ul style="list-style-type: none"><li>• Michael L. Scott: Programming Language Pragmatics, Morgan Kaufmann, 2016</li><li>• Keith D. Cooper, Linda Torczon: Engineering a Compiler, 2nd edition, Morgan Kaufman, 2011</li><li>• Dick Grune et. al.: Modern Compiler Design, Springer, 2016</li></ul>
Leistungsnachweis	Schriftliche Prüfung (90 min)



# Software Engineering

Studiengang	Bachelor Informatik 2. Studienabschnitt Softwaremethoden und -technologien
Modul	Software Engineering
Modulverantwortliche	Bartosz von Rymon Lipinski, Matthias Meitner
Dozent(en)	Matthias Meitner, Bartosz von Rymon Lipinski
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"><li>• Probleme der industriellen Softwareerstellung;</li><li>• Phasenmodelle;</li><li>• Methoden zur Anforderungsspezifikation; Entwurfsmethoden;</li><li>• Methoden zur Systemkonstruktion; Systemintegration und Test; Software-Ergonomie;</li><li>• Qualitätssicherung; Softwaremetriken; Projektmanagement; DV-gestützte Entwicklungsumgebungen;</li><li>• Aufwandsschätzungen; Wirtschaftlichkeitsbetrachtungen.</li></ul>
Literatur	<ul style="list-style-type: none"><li>• C. Rupp, S. Queins, UML2 glasklar, 2012, Hanser.</li><li>• G. Pomberger, W. Pree, Software Engineering, München, 2004, Hanser.</li><li>• I. Sommerville, Software Engineering, München, 2012, Pearson Studium.</li></ul>
Leistungsnachweis	Informatik und Wirtschaftsinformatik: Schriftliche Prüfung (90 min). Medieninformatik: Portfolioprüfung
Zulassungsvoraussetzung	

## Software Engineering Praktikum

Studiengang	Bachelor Informatik 2. Studienabschnitt Softwaremethoden und -technologien
Modul	Software Engineering Praktikum
Modulverantwortliche	Bartosz von Rymon Lipinski, Birgit Kraft, Matthias Meitner
Dozent(en)	Thomas Chang, Birgit Kraft, Simon Seibt
Leistungspunkte	2
Semesterwochenstunden	2
Semesterturnus	Winter- bzw. Sommersemester
Unterrichtssprache	Kurs nur in Deutsch

# Softwarearchitektur

Studiengang	Bachelor Informatik 2. Studienabschnitt Softwaremethoden und -technologien
Modul	Softwarearchitektur
Modulverantwortliche	Ramin Tavakoli Kolagari, Korbinian Riedhammer
Dozent(en)	Ramin Tavakoli Kolagari
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 in Deutsch und Englisch
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"><li>• Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: "Design Patterns, Elements of Reusable Object-Oriented Software", Addison Wesley, 1994.</li><li>• Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal: "Pattern-Oriented Software Architecture", Wiley, 1996.</li></ul>

- 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)

Zulassungsvoraussetzung

## Betriebssysteme

Studiengang	Bachelor Informatik 2. Studienabschnitt Systeminfrastrukturen
Modul	Betriebssysteme
Modulverantwortliche	Friedhelm Stappert, Michael Zapf
Dozent(en)	Friedhelm Stappert, 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"><li>• Einsicht in die theoretischen Grundlagen von Betriebssystemen; exemplarisch verdeutlicht an Hand konkreter Beispiele.</li><li>• Aufbau und Architektur von Betriebssystemen verstehen und analysieren.</li><li>• Einfache, betriebssystemspezifische Programme erstellen können (Systemprogrammierung).</li><li>• Parallele Prozesse und Synchronisationsmechanismen verstehen und anwenden.</li><li>• Fähigkeit, aktuelle Betriebssysteme und ihre Dienste einzuordnen und zu bewerten.</li></ul>
Schlüsselqualifikation	Fähigkeit zu komplexem Denken, Ausdauer bei Problemlösungen.
Lehrinhalte	<ul style="list-style-type: none"><li>• Aufgaben und Aufbau von Betriebssystemen</li><li>• Verwaltungsfunktionen (insbesondere Prozesse, Speicher, Dateien)</li><li>• Kommunikationsfunktionen</li></ul>
Literatur	<ul style="list-style-type: none"><li>• Eduard Glatz: Betriebssysteme – Grundlagen, Konzepte, Systemprogrammierung, dpunkt.verlag, 3. Auflage, Heidelberg, 2015</li><li>• Andrew S. Tanenbaum: Moderne Betriebssysteme, 2016, Pearson</li><li>• William Stallings: Operating Systems, Prentice Hall, 2008</li><li>• M. Mitchell, J. Oldham, A. Samuel: Advanced Linux Programming. New Riders, 2001</li><li>• Peter Mandl: Grundkurs Betriebssysteme. Springer 2013</li><li>• Vorlesungsunterlagen</li></ul>
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 Kontaktzeit, 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"><li>• Peterson L., Davie B., Computer Networks: A Systems Approach, 4th Edition, 2008, Morgan Kaufmann Publishers. (deutsche Übersetzung "Computernetze" erschienen im dPunkt Verlag, Heidelberg, 2008)</li><li>• Andrew S. Tanenbaum, Computer Networks, 4th Edition, Upper Saddle River, NJ, 2002, Prentice Hall PTR (auch deutsche Übersetzung erschienen bei Pearson Studium, 2003)</li><li>• Roth J., Prüfungstrainer Rechnernetze, Vieweg+Teubner, 2010</li></ul>
Leistungsnachweis	Schriftliche Prüfung (90 min)
Zulassungsvoraussetzung	

## Grundlagen der Wirtschaftswissenschaften

Studiengang	Bachelor Informatik 2. Studienabschnitt Vertiefende Wahlpflichtfächer
Modul	Grundlagen der Wirtschaftswissenschaften
Modulverantwortliche	Patricia Brockmann, Heidi Schuhbauer, Michael Lang
Dozent(en)	Patricia Brockmann, Michael Lang, Heidi Schuhbauer
Vorkenntnisse	Schulwissen
Arbeitsaufwand	210 Stunden, davon: 90 Stunden Präsenz, 120 Stunden Vor- und Nachbereiten des Lehrstoffs
Leistungspunkte	7
Semesterwochenstunden	6
Veranstaltungstyp	6 SWS Vorlesung Vorlesung mit integrierten Übungen
Semesterturnus	Wintersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Betriebswirtschaftliche und volkswirtschaftliche Grundlagen
Lernziel	Grundkenntnisse über Aufbau und Funktionen von Betrieben Überblick über grundlegende volkswirtschaftliche Zusammenhänge
Schlüsselqualifikation	Denken in wirtschaftlichen Zusammenhängen
Lehrinhalte	Wirtschaftswissenschaftliche Grundbegriffe; Aufbau und Funktionen von Betrieben (Standortwahl, Rechtsformen, Produktionsfaktoren); Grundbegriffe des betrieblichen Rechnungswesens. Marktformen und Preisbildung, Arbeitsteilung und Arbeitsmarkt, Distributions-, Allokations- und Stabilisierungsfunktion des Staates, Geld- und Fiskalpolitik, gesamtwirtschaftliches Gleichgewicht
Literatur	Straub, T.: Einführung in die Allgemeine Betriebswirtschaftslehre, Pearson. Bofinger, P.: Grundzüge der Volkswirtschaftslehre, Person.
Leistungsnachweis	Schriftliche Prüfung (90 min)

# Informationssicherheit und ihre Grundlagen

Studiengang	Bachelor Informatik 2. Studienabschnitt Vertiefende Wahlpflichtfächer
Modul	Informationssicherheit und ihre Grundlagen
Modulverantwortliche	Ronald Petrlc
Dozent(en)	Ronald Petrlc
Vorkenntnisse	Lehrveranstaltungen des ersten Studienjahres, insbesondere Mathematik I,II, Grundlagen der Wirtschaftsinformatik, 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 Wirtschaftsinformatiker 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"><li>• Ziele in der Informationssicherheit: Vertraulichkeit, Datenintegrität, Authentifizierung des Ursprungs von Daten, Authentifizierung von Entitäten, Verbindlichkeit;</li><li>• Symmetrische Verschlüsselungsverfahren: Stromchiffre, Blockchiffre (z. B. DES, AES), Operationsmodi;</li><li>• Kryptographische Hashfunktionen, Message Authentication Codes (MAC);</li><li>• Asymmetrische Verfahren / Public Key - Kryptographie: Einwegfunktionen, Verschlüsselung, digitale Signatur, RSA-, ElGamal/DSA-Verfahren; Angriffe; optimal asymmetric encryption padding OAEP; Kryptographische Protokolle für den Schlüsselaustausch;</li></ul>



- 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, 3<sup>rd</sup> 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).

# Kryptographie und Informationssicherheit

Studiengang	Bachelor Informatik 2. Studienabschnitt Vertiefende Wahlpflichtfächer
Modul	Kryptographie und Informationssicherheit
Modulverantwortliche	Hans Löhr
Dozent(en)	Hans Löhr
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"><li>• Ziele in der Informationssicherheit: Vertraulichkeit, Datenintegrität, Authentifizierung des Ursprungs von Daten, Authentifizierung von Entitäten, Verbindlichkeit;</li><li>• Symmetrische Verschlüsselungsverfahren: Stromchiffre, Blockchiffre (z. B. DES, AES), Operationsmodi;</li><li>• Kryptographische Hashfunktionen, Message Authentication Codes (MAC);</li><li>• Asymmetrische Verfahren / Public Key - Kryptographie: Einwegfunktionen, Verschlüsselung, digitale Signatur, RSA-, ElGamal/DSA-Verfahren; Angriffe; optimal asymmetric encryption padding OAEP; Kryptographische Protokolle für den Schlüsselaustausch;</li></ul>

- 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, 3<sup>rd</sup> 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).

# Machine Learning

Studiengang	Bachelor Informatik 2. Studienabschnitt Vertiefende Wahlpflichtfächer
Modul	Machine Learning
Modulverantwortliche	Tobias Bocklet
Dozent(en)	Tobias Bocklet
Vorkenntnisse	
Arbeitsaufwand	210 Stunden, davon 90 Präsenz und 120 Selbststudium
Leistungspunkte	7
Semesterwochenstunden	6
Veranstaltungstyp	4 SWS Vorlesung, 2 SWS Übung
Semesterturnus	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"><li>• Grundlegende Prinzipien und Theorien des Maschinellen Lernens</li><li>• Einführung der wichtigsten algorithmischen Konzepte</li><li>• Lernen aus Daten</li><li>• Praktische Anwendungen</li></ul>
Literatur	<ul style="list-style-type: none"><li>• Duda, R and Hart, P and Stork, D: Pattern Classification, 2. Auflage, 2001, Wiley, NY</li><li>• Geron A: HANds-On Machine Learning with Scikit-Learn, Keras &amp; TensorFlow, 2. Auflage, 2019, O'Reilly Media</li><li>• Niemann,H: <i>Klassifikation von Mustern</i>. 2. Überarbeitete Auflage, 2003</li><li>• Goodfellow, I and Bengio,Y and Courville, A: <i>Deep Learning</i>. 2016</li></ul>
Leistungsnachweis	Schriftliche Prüfung (90min)
Zulassungsvoraussetzung	

# Mathematische Methoden für maschinelles Lernen

Studiengang	Bachelor Informatik 2. Studienabschnitt Vertiefende Wahlpflichtfächer
Modul	Mathematische Methoden für maschinelles Lernen
Modulverantwortliche	Elke Wilczok, Matthias Börger, Alexander Hufnagel
Dozent(en)	Jürgen Bolik, Alexander Hufnagel, Elke Wilczok
Vorkenntnisse	Mathematik 1 und 2; Grundlagen Statistik (für Teil 2)
Arbeitsaufwand	150 Stunden, davon: 50 Stunden Präsenzzeit 100 Stunden Vor- und Nachbereitung der Vorlesung, des Vortrags zu einer Fallstudie und Abschlussprüfung
Leistungspunkte	7
Semesterwochenstunden	6
Veranstaltungstyp	Vorlesung mit Übungen
Semesterturnus	Winter- und Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Lernziel	Nach der Teilnahme an diesem Modul sind die Studierenden in der Lage <ul style="list-style-type: none"><li>• weiterführende Module im Bereich KI bzw. maschinelles Lernen ohne Behandlung der mathematischen Grundlagen zu besuchen</li><li>• ausgewählte mathematische Anwendungen auf konkrete Fallbeispiele umzusetzen.</li></ul>
Schlüsselqualifikation	<ul style="list-style-type: none"><li>• Problemlösen (Analyse und Klassifikation von Problemen)</li><li>• Denkweisen (abstraktes und logisches Denken)</li></ul>
Lehrinhalte	Lehrinhalte Teil 1 (Winter) <ul style="list-style-type: none"><li>• Erweiterte Kenntnisse in Linearer Algebra und Analysis: Matrixzerlegungen, Tensoren, evtl. etwas Numerik</li><li>• Nichtlineare Optimierung: Gradientenverfahren, Newton- und Quasi-Newton-Verfahren</li><li>• Funktionenräume und Approximation in Funktionenräumen</li><li>• Regression: Minimierung quadratischer Fehler (tw. Wiederholung, evtl. im Sommer)</li><li>• Fortsetzung Mehrdimensionale Analysis: Kettenregel, Automatische Differentiation</li><li>• Grundkenntnisse gewöhnliche Differentialgleichungen</li></ul> Lehrinhalte Teil 2 (Sommer) <ul style="list-style-type: none"><li>• Wiederholung Statistik (ML-Schätzer, Verteilungen)</li><li>• Einführung in die Bayes-Statistik</li><li>• Grundkenntnisse in stochastischen Prozessen, insbesondere Markov-Prozessen</li><li>• Mathematische Verfahren der Signalverarbeitung (diskrete Fouriertransformation, Datenkompression)</li><li>• Grundkenntnisse und elementare Begriffe in mehrdimensionaler Integration (evtl. im Winter, kurz)</li><li>• Konzepte der mathematischen Optimierung: Dynamisches Programmieren, heuristische Suchverfahren (z.B. evolutionäre Algorithmen)</li></ul>

	<ul style="list-style-type: none"> <li>• Informationstheorie, Begriff Entropie</li> </ul>
Literatur	<ul style="list-style-type: none"> <li>• Aggarwal, C.C., Linear Algebra and Optimization for Machine Learning, Springer International Publishing; 2020</li> <li>• Deisenroth, M.P., Faisal, A.A., and Ong, C.S., Mathematics for Machine Learning, Cambridge University Press; 2020</li> </ul>
Leistungsnachweis	Schriftliche Prüfungen (Modulteilprüfungen) für Winter und Sommer je 60 Minuten
Zulassungsvoraussetzung	Mathematik 1 und 2 (für Kurs Winter), für den Sommer noch Statistik.

# Mensch-Computer-Interaktion

Studiengang	Bachelor Informatik 2. Studienabschnitt Vertiefende Wahlpflichtfächer
Modul	Mensch-Computer-Interaktion
Modulverantwortliche	Alexander Kröner, Timo Götzelmann
Dozent(en)	Timo Götzelmann, Alexander Kröner, Roman Stöhr
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 nur in Deutsch
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"><li>• Historische Entwicklung und Arten von Benutzerschnittstellen</li><li>• Interaktionstechnologien und deren Interaktionstechniken</li><li>• Graphische Dialogsysteme</li><li>• Modelle und Architekturmuster für interaktive Systeme</li><li>• Entwicklungsframeworks</li></ul>
Leistungsnachweis	Schriftliche Prüfung (90 min.)

# Natural Language Processing

Studiengang	Bachelor Informatik 2. Studienabschnitt Vertiefende Wahlpflichtfächer
Modul	Natural Language Processing
Modulverantwortliche	Jens Albrecht, Korbinian Riedhammer, Anna Kruspe
Dozent(en)	Jens Albrecht
Vorkenntnisse	Keine Wünschenswert: Python, Grundlagen Machine Learning
Arbeitsaufwand	210 Stunden, davon:  90 Stunden Präsenzzeit 120 Stunden Vor- und Nachbereitung der Vorlesungen und Übungen sowie Bearbeitung des Seminarprojekts
Leistungspunkte	7
Semesterwochenstunden	6
Veranstaltungstyp	2 SWS Vorlesung zur Vermittlung notwendiger Grundlagen 2 SWS Übung zu den gelehrten Methoden anhand von Jupyter-Notebooks 2 SWS Teamarbeit an selbstgewählten Seminarprojekten + Präsentation
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs in Deutsch und/oder Englisch (bitte nachfragen)
Beitrag zu den Zielen des Studiums	Verstehen, Anwendung und eigenes Entwerfen komplexer Algorithmen
Lernziel	Nach der Teilnahme an diesem Modul sind die Studierenden in der Lage <ul style="list-style-type: none"><li>• Grundlegende Begriffe und Konzepte des Natural Language Processing zu verstehen</li><li>• Methoden zur Lösung verschiedener Aufgaben in diesem Bereich zu implementieren</li><li>• Lösungen für praktische Anwendungen der Textanalyse zu entwerfen und verschiedene Alternativen zu bewerten</li></ul>
Schlüsselqualifikation	<ul style="list-style-type: none"><li>• Problemlösen (Analyse und Klassifikation von Problemen)</li><li>• Denkweisen (abstraktes und logisches Denken),</li><li>• Umgang mit wissenschaftlicher Literatur (englischsprachig),</li><li>• Teamfähigkeit,</li><li>• Schreiben (wissenschaftliches Schreiben) und</li><li>• Vortragen (Präsentationsfähigkeiten, Fachvorträge).</li></ul>
Lehrinhalte	In vielen Domänen liegt der Großteil der Information in Form unstrukturierter Textdaten in natürlicher Sprache vor, insbesondere mit Hinblick auf die riesigen im Internet verfügbaren Datenmengen. Hierzu zählen beispielsweise Zeitungsartikel, Social Media-Daten und Nutzerkommentare, Fachartikel, künstlerische Texte usw. Die große Menge dieser Daten erfordert automatisierte Analysemethoden, welche bereits vielfältig eingesetzt werden. Ziel dieser Veranstaltung ist es, die Studierenden mit den grundlegenden Aspekten der Textverarbeitung sowie Methoden des Natural Language Processing vertraut zu machen. Zu den Inhalten zählen z.B.:

## Grundlagen



- Linguistische Grundlagen
- Textcorpora
- Textvorverarbeitung, Reguläre Ausdrücke, Distanzmaße
- Tokenization
- Language Models
- Naive Bayes- und MaxEnt-Methoden
- Repräsentationen und Embeddings
- Neuronale Netze für Text

### Methoden

- Part-of-speech tagging
- Named Entity Recognition
- Constituency/dependency parsing
- Word sense disambiguation
- Topic modeling
- Semantic Role Labeling
- Coreference resolution

### Anwendungen

- Wortprädiktion, Autokorrektur
- Textklassifikation
- Summarization
- Question Answering, Information Retrieval
- Sentiment analysis
- Machine Translation
- Text generation
- Dialogsysteme

### Literatur

- Jurafsky, D., & Martin, J.H. (2021). Speech and language processing. <https://web.stanford.edu/~jurafsky/slp3/>
- Albrecht, J., Ramachandran, S., Winkler, C. (2020). Blueprints for Text Analytics Using Python. O'Reilly, 2020. <https://learning.oreilly.com/library/view/-/9781492074076/?ar>
- Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. <https://www.nltk.org/book/>
- Vasiliev, Y. (2020). Natural Language Processing with Python and spaCy. No starch press.
- [Manning, C., & Schütze, H. (1999). Foundations of Statistical Natural Language Processing. MIT Press.]

### Leistungsnachweis

Studienarbeit in Form eines Abschlussprojekts mit Vortrag und mündliche Befragung (15 min). Gewichtung 60% Projekt, 40% Befragung.

### Zulassungsvoraussetzung

## Webanwendungen

Studiengang	Bachelor Informatik 2. Studienabschnitt Vertiefende Wahlpflichtfächer
Modul	Webanwendungen
Modulverantwortliche	Matthias Teßmann
Dozent(en)	Matthias Teßmann
Vorkenntnisse	Programmieren I und II, Software Engineering, Grundlagen der Informatik
Arbeitsaufwand	210 Stunden, davon: 90 Stunden Kontaktzeit, 120 Stunden Vor- und Nachbereiten des Lehrstoffs sowie Übungsaufgaben/Praktikum
Leistungspunkte	7
Semesterwochenstunden	6
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"><li>• Grundlagen<ul style="list-style-type: none"><li>◦ HTML5, CSS3, JavaScript</li></ul></li><li>• Client-Server Modell / Kommunikationsprotokolle und Datenformate<ul style="list-style-type: none"><li>◦ HTTP/HTTPS/HTTP2</li><li>◦ JSON</li></ul></li><li>• Front-End, Back-End und "Full-Stack"-Anwendungsentwicklung</li></ul>

- 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, Reihnwerk 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

Zulassungsvoraussetzung

## 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)	Wolfgang Bremer, Florian Gallwitz, Anna Kruspe, Ronald Petrlc, Joachim Scheja, Heidi Schuhbauer, Peter Trommler
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"><li>• Präsentationstechniken</li><li>• Feedback und Diskussion</li><li>• Gestaltung von Berichten</li></ul>
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	Martin Geier, 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"><li>• Projektplanung, -steuerung und -kontrolle mit Prozessmodellwahl</li><li>• Meilensteinplanung</li><li>• Werte, Prinzipien und Vorgehensmodelle des agilen Projektmanagements</li><li>• Terminplanung mit Netzplantechnik</li><li>• Kosten- und Aufwandsschätzung</li><li>• Ressourcenzuordnung</li><li>• Steuerung und Überwachung</li><li>• Dokumentation</li><li>• Überprüfung der Zielerreichung</li><li>• Lernen aus Projekten</li></ul>
Literatur	<ul style="list-style-type: none"><li>• Hindel, B. u. a.: Basiswissen Projektmanagement, Heidelberg 2004, dpunkt.Verlag</li><li>• Tumascheit, K. D.: Überleben im Projekt. 10 Projektfallen und wie man sie umgeht, München (Redline) 2014</li><li>• Heintel, P.; Krainz, E. E.: Projektmanagement. Hierarchiekriese. Systemabwehr. Komplexitätsbewältigung, 5. Auflage, Wiesbaden (Gabler) 2011</li><li>• Schelle, H.; Ottmann, R.; Pfeiffer, A.: ProjektManager, 3. Auflage, Nürnberg (Deutsche Gesellschaft für Projektmanagement) 2008</li></ul>
Leistungsnachweis	Dieser Leistungsnachweis kann nur praxisbegleitend abgelegt werden. (Klausur als digitale Prüfung im Prüfungssystem Exams 90 Min.)

Zulassungsvoraussetzung

## Aktuelle Entwicklungen des Computer Designs

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Aktuelle Entwicklungen des Computer Designs
Modulverantwortliche	Axel Hein
Dozent(en)	Axel Hein
Vorkenntnisse	Inhaltliche Voraussetzung ist der erfolgreiche Besuch der Lehrveranstaltungen Computerarchitektur und Betriebssysteme, wie sie verpflichtend im Studiengang BA IN angeboten werden.
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Anwesenheit, 85 Stunden Bearbeitung der Vortrags- und Projektthemen.
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Seminar
Semesterturnus	Winter- bzw. Sommersemester
Unterrichtssprache	Kurs in Deutsch und/oder Englisch (bitte nachfragen)
Beitrag zu den Zielen des Studiums	Kennenlernen, Verstehen und Bewertung von architekturellen Konzepten und Vorstellung typischer Beispiele mit Vertiefung und Erweiterung der Grundlagen aus den Lehrveranstaltungen Computerarchitektur und Betriebssysteme.
Lernziel	Nach erfolgreicher Teilnahme verfügen die Studierenden über vertiefte Kenntnisse in aktuellen Entwicklungen der Technologie, Hardware-Architektur und Betriebssystem-Aspekte in aktuellen Computersystemen.
Schlüsselqualifikation	Fähigkeit zu komplexem Denken, Ausdauer bei Problemlösungen.
Lehrinhalte	Aktuelle Entwicklungen im Design, in der verwendeten Hardware und in Betriebssystemen von Computersystemen werden konzeptuell sowie anhand aktueller Fallstudien diskutiert. Themen der Lehrveranstaltung sind aktuelle Entwicklungen und konkrete Umsetzungen bei der verwendeten Technologie sowie bei architekturellen Aspekten. Dies betrifft unter anderem spezielle Strukturen in Prozessor-ICs, ASICs, Speicher-ICs, Flash-ICs, FPGAs, sowie Design-Entscheidungen wie auch Realisierungen bei MultiCore-Architekturen mit ihren IC-internen Speicherhierarchien und Verbindungsstrukturen, Caching, Pipelining, Superskalarität, Parallelität, Speichermedien sowie deren Leistungsfähigkeit, Robustheit und Lebensdauer, spezielle Prozessor-Realisierungen (z.B. GPU - Graphics Processing Unit, TPU - Tensor Processing Unit), Dateisysteme auf verschiedenen Speichermedien, Fehlertoleranz sowie weitere technologische und architekturelle Aspekte der Steigerung der Leistungsfähigkeit und Zuverlässigkeit von Computersystemen.
Literatur	Siehe die in der Lehrveranstaltung diskutierten Themen.
Leistungsnachweis	Anwesenheitspflicht Seminar (Vortrag & Ausarbeitung)
Zulassungsvoraussetzung	Inhaltliche Voraussetzung ist der erfolgreiche Besuch der Lehrveranstaltungen



Computerarchitektur und Betriebssysteme, wie sie verpflichtend im Studiengang BA IN angeboten werden.

# 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"><li>• Ausfallsicherheit von Netzen</li><li>• Suchstrategien</li><li>• Finden von kürzesten Wegen</li><li>• Routenplanung</li><li>• Zuordnungsprobleme</li><li>• Maximale Flüsse in Netzwerken</li></ul> <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"><li>• Christina Büsing, Graphen- und Netzwerkoptimierung, Spektrum 2010</li><li>• Shimon Even, Graph Algorithms, 2nd ed., Cambridge 2012</li><li>• Dieter Jungnickel, Graphs, Networks and Algorithms, 3rd ed., Springer 2007</li><li>• Sven Oliver Krumke, Hartmut Noltemeier: Graphentheoretische Konzepte und Algorithmen, 3. Auflage, Springer 2012</li><li>• Angelika Steger, Diskrete Strukturen 1, Springer Verlag 2007</li><li>• Volker Turau, Algorithmische Graphentheorie, Oldenbourg 2009</li><li>• T. Cormen et al: Introduction to Algorithms, MIT Press, 2009</li></ul>
Leistungsnachweis	Schriftliche Prüfung (90 min)

# Angewandtes Projektmanagement

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Angewandtes Projektmanagement
Modulverantwortliche	Carsten Roßleben
Vorkenntnisse	keine
Arbeitsaufwand	150 Stunden, davon: 60 Stunden Präsenz und 90 Stunden Vor- und Nachbereitung
Leistungspunkte	5
Semesterwochenstunden	4
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Systematisches Vorgehen im Projektmanagement im Zusammenhang mit verschiedenen Softwarelösungen
Lernziel	Nach der erfolgreichen Teilnahme an diesem Modul kennen die Studierenden die fachlichen und kommunikativen Herausforderungen, die im angewandten Projektmanagement begründet sind. Die Studierenden können die Potenziale und Grenzen aktueller Tools, Konzepte und Analyse-Verfahren im Projektmanagement-Umfeld beschreiben und sind dadurch in der Lage, Anwendungsfälle systematisch zu analysieren und Lösungsansätze zu entwickeln.
Schlüsselqualifikation	Durchführung von Projekten, Voraussetzungen für Projekte, Verständnis von Rollen in Projekten, Problemanalyse, agile Methoden, Projektabnahme, PM-Tools, Kommunikation während des Projektes
Lehrinhalte	An 5 Fallstudien und einem eigenen Projekt werden Methoden des Projektmanagement anhand verschiedener Softwarelösungen erarbeitet <ul style="list-style-type: none"><li>- Prozesse, Prozessmodelle, Methoden, Begriffe</li><li>- Projektorganisation, Projektplanung, Projektbewertung</li><li>- Konflikte beim PM</li><li>- Phasenmodell</li><li>- Die Projektleitung</li><li>- Das Projektteam</li><li>- EDV-Unterstützung für das Projektmanagement</li><li>- Projekt-Controlling</li></ul>
Leistungsnachweis	Selbständiges Erarbeiten von diversen Lösungen im Rahmen von Fallstudien und selbständiges planen und bearbeiten eines eigenen Projektes.

## Autonome mobile Roboter

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Autonome mobile Roboter
Modulverantwortliche	Jörg Roth
Vorkenntnisse	Programmieren, Grundlagen der Informatik
Arbeitsaufwand	150 Stunden, davon: 60 Stunden Kontaktzeit, 70 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 70 Stunden, mündliche Befragung (ca. 25 min), jeweils gewichtet mit 50%



## Betriebssysteme für einfache Mikrocontroller

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Betriebssysteme für einfache Mikrocontroller
Modulverantwortliche	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"><li>• 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.</li><li>• 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.</li><li>• Die AbsolventInnen dieser Lehrveranstaltung können unter Einbindung eines einfachen Betriebssystems Anwendungen für Mikrocontroller planen und implementieren.</li></ul>
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"><li>• Wie laufen Programme in Vorder-/Hintergrundsystemen ab?</li><li>• Was bestimmt das Zeitverhalten?</li><li>• Welchen Einfluss haben Interrupts?</li><li>• Welchen Vorteil bietet die Verwendung eines Betriebssystems?</li><li>• Wie unterscheidet sich das Software Design?</li></ul>

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](http://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
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"><li>• Übersicht über die Grundlagen der Software-Qualität</li><li>• Softwaremetriken, Metrikanwendung in der Praxis</li><li>• Überblick über Prinzipien, Best Practices und Code Smells</li><li>• Einhaltung und Überprüfung Java Code Conventions</li><li>• Statische Softwareprüfung, insbesondere Review-Techniken und statische Programmanalyse</li><li>• Sicherung der Softwarequalität mit Werkzeugen wie SonarQube, PMD, SpotBugs und Checkstyle</li><li>• Softwaretests mit JUnit</li><li>• Überprüfen der Testabdeckung (Code Coverage)</li><li>• Continuous Integration</li><li>• Design Prinzipien</li><li>• Design Patterns (GoF)</li></ul>
Literatur	<ul style="list-style-type: none"><li>• Schneider, Kurt: Abenteuer Software Qualität – Grundlagen und Verfahren für Qualitätssicherung und Qualitätsmanagement, dpunkt.verlag, 2007</li><li>• Robert, Martin: Clean Code – Refactoring, Patterns, Testen und Techniken für sauberen Code, mitp-Verlag, 2009</li></ul>



- 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)

Zulassungsvoraussetzung

# Computational Intelligence

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Computational Intelligence
Modulverantwortliche	Alaa Sheta
Vorkenntnisse	none
Arbeitsaufwand	150 hours, including:  65 hours attendance,  85 hours preparation and revision
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Block seminar
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs nur in Englisch
Beitrag zu den Zielen des Studiums	Application and deepening knowledge in the field of computational intelligence.
Lernziel	On concluding the course, candidates will have an in-depth knowledge of theories, methods, and algorithms in computational intelligence. Candidates will be able to apply the apply these methods on solving various optimization problems in real-life applications.
Schlüsselqualifikation	Key Qualifications:  1. Explore various software tools for building computational intelligence systems.  2. Understand the basic idea of system modeling/function optimization.  3. The student will get a hand experience with several methods to handle the data set for modeling problems such as data pre-processing and how to split the data training and testing dataset.  4. Understand how we can build a simple regression model with single and multiple variables; this is a common technique to solve various prediction problems.  5. Understand how the natural selection mechanism inspires evolutionary computation algorithms.  6. Understand the meaning of search space/landscape and difficulties associated with searching for the minimum or maximum of a function.  7. Understand the meaning of fitness/evaluation function values that guide a search problem.  8. Learn how to formulate a real-life problem such that a computational intelligence algorithm can be used to solve it.

9. The student will learn how to evaluate several computational intelligence algorithms' strengths and weaknesses and understand which algorithm is suitable for which problem.
10. Understand the benefits and weaknesses of Artificial Neural Networks (ANN) in solving various function approximation problems.
11. Learn how fuzzy logic can approximate a nonlinear/complex function-based set of linear models and how the problem domain can be decomposed to different sub-domain using membership functions.
12. Learn to assess and understand the key challenges in using CI in applications such as manufacturing process modeling and system identification.

#### Lehrinhalte

This course provides a broad introduction to computational intelligence techniques.

Computational Intelligence (CI) is the theory, design, application, and development of biologically and linguistically motivated computational paradigms. Traditionally the three main pillars of CI have been Neural Networks, Fuzzy Systems, and Evolutionary Computation. The course will also discuss recent applications of CI in health care, manufacturing process modeling, autonomous robot navigation, speech and image processing, and text and web data processing.

The course is intended for students interested in learning more about how intelligent systems work and their capacity to solve complex real-world problems through nature-inspired algorithms. The course will focus on the main CI approaches and methodologies, such as genetic algorithms, swarm optimization, artificial neural networks, and fuzzy systems.

#### Literatur

- James M. Keller, Derong Liu, David B. Fogel, Fundamentals of Computational Intelligence: Neural Networks, Fuzzy Systems, and Evolutionary Computation (IEEE Press Series on Computational Intelligence) 1st Edition, 2016.
- Sumathi, Surekha Paneerselvam, Computational Intelligence Paradigms: Theory & Applications using MATLAB by SCRC Press, 2010.
- Andries P. Engelbrecht, Computational Intelligence: An Introduction, 2nd Edition, ISBN: 978-0-470-03561-0, November 2007.
- Y. Daniel Liang, Introduction to Programming with Python, Pearson, 2013.
- Stormy Attaway, Matlab: A Practical Introduction to Programming and Problem Solving, Elsevier, 2009.

#### Leistungsnachweis

Presentation (25 min) 25%

Written paper 25%

Paper-based Exam (45 min) 50%

## Datenfusion

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Datenfusion
Modulverantwortliche	Cristian Axenie
Dozent(en)	Cristian Axenie
Vorkenntnisse	Mathematik, Programmierung
Arbeitsaufwand	150h, davon: 65h Präsenz, 85h Nachbereitung des Lehrstoffs und Prüfungsvorbereitung
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Vorlesung mit Übungen/Labor
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs in Deutsch und Englisch
Beitrag zu den Zielen des Studiums	Die Datenfusion ist ein grundlegendes Verfahren in vielen wissenschaftlichen und technischen Anwendungen. Sie ist definiert als ein mehrstufiger Prozess, der sich auf die Assoziation, Korrelation und Kombination von Daten und Informationen aus verschiedenen Quellen konzentriert. Ziel ist es, eine höhere Genauigkeit und bessere Kenntnisse zu gewinnen, als dies durch die Verwendung eines einzelnen Sensors allein möglich wäre. In dieser Lehrveranstaltung analysieren und diskutieren wir multisensorische Fusionstechniken, Architekturen und funktionale Aspekte der Datenassoziation, -korrelation und -kombination. Der theoretische Hintergrund wird durch praktische Beispiele und Anwendungen unterstützt. Die praktischen Anwendungen konzentrieren sich auf ressourcenlimitierte Geräte wie Mikrocontroller oder Edge Devices.
Lernziel	Das Erlernen der Motivation und des Nutzens der Sensorfusion, der Zuverlässigkeit von Daten, der Korrelation von Daten und schließlich der Kombination von Daten. Das gesamte Spektrum der Ansätze von der Signalverarbeitung bis hin zu neuronalen Netzen wird vergleichend beschrieben. Es werden einige Tipps und Tricks für den praktischen Einsatz der Algorithmen zur Datenfusion gegeben.
Schlüsselqualifikation	Die Implementierung (mathematische Beschreibung und Programmierung) von Datenfusionsalgorithmen für 1Dimensionale Daten (Zeitreihen) und auch höherdimensionale Daten (Bilddaten). Die Bewertung von Datenfusionsalgorithmen.
Lehrinhalte	<p>In der Vorlesung werden schrittweise (komplexe) Algorithmen für die Datenfusion sowohl in niedrigen als auch in hohen Dimensionen vorgestellt. Die Algorithmen werden mathematisch beschrieben und praktisch in Software implementiert. Die praktischen Anwendungen konzentrieren sich auf ressourcenlimitierte Geräte wie Mikrocontroller oder Edge Devices.</p> <p>Teil 1: Daten Fusion Klassifikation, Funktionalität, Architekturen</p> <ul style="list-style-type: none"><li>• Monomodale und multimodale Datenfusion</li><li>• Zentralisierte und dezentralisierte Datenfusion</li></ul>

- Datenfusion auf Signalebene
- Datenfusion auf Merkmalsebene
- Datenfusion auf Entscheidungsebene
- Lokale und globale Datenfusion

## Teil 2: Gängige Verfahren und Stand der Technik

- Signalverarbeitungsmethoden zur Datenfusion
- Lernende Datenfusionssysteme

### Literatur

- Mitchell, Harvey B. Multi-sensor data fusion: an introduction. Springer Science & Business Media, 2007.
- Hall, David Lee, and Sonya AH McMullen. Mathematical techniques in multisensor data fusion. Artech House, 2004.
- Liggins II, Martin, David Hall, and James Llinas, eds. Handbook of multisensor data fusion: theory and practice. CRC press, 2017.

### Leistungsnachweis

Praktisches Software (Team-)Projekt und abschließende 10-minütige Präsentation.

### Zulassungsvoraussetzung

## Datenmodellierung mit XML

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Datenmodellierung mit XML
Modulverantwortliche	Alexander Kröner
Vorkenntnisse	Erforderlich  <ul style="list-style-type: none"><li>• Programmieren I, II</li><li>• Web-Programmierung (insbesondere HTML, PHP)</li></ul> Empfohlen  <ul style="list-style-type: none"><li>• Javascript (Grundkenntnisse)</li><li>• ERM</li></ul>
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenzzeit 85 Stunden Vor- und Nachbereitung, Prüfungsvorbereitung
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Vorlesung, Übungen
Semesterturnus	Wintersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Die Beherrschung der weit verbreiteten XML-Technologien trägt zur praxisorientierten Ausbildung bei. Die für das Fach charakteristischen semi-strukturierte Datenmodelle sind ein Bestandteil moderner Datenbankkonzepte (NoSQL). Der Umgang mit XML Schema und XSLT ermöglicht einen Einblick in Grammatiken und regelbasierte Systeme als Komponenten fortgeschrittener Programmiersprachen.
Lernziel	Kenntnis der Eigenschaften von semi-strukturierten Datenmodellen und well-formed XML. Verstehen der Auswirkungen dieses Modellierungsansatzes. Anwendung dieses Wissens um konzeptionelle Datenmodelle zu analysieren und ein korrespondierendes Model mittels XML Schema zu entwickeln. Fähigkeit, XSLT anzuwenden und zu entwickeln als Werkzeug zur Transformation von Modellen.
Schlüsselqualifikation	Theoretisches Verständnis und praktische Beherrschung ausgewählter XML-Technologien.
Lehrinhalte	Die Extensible Markup Language (XML) ist eine Auszeichnungssprache zur Repräsentation semi-strukturierter Daten in Form von Textdateien. XML bildet im Internet eine der Basistechnologien für den plattform- und implementationsunabhängigen Austausch von Daten. Schwerpunkt der Vorlesung bilden u.a.:  <ul style="list-style-type: none"><li>• Grundlagen semi-strukturierter Datenmodelle: Strategien zur Serialisierung relationaler Modelle, well-formed XML (u.a. Namenskonventionen, Whitespace-</li></ul>

Behandlung, Namensräume)

- Dokumentstrukturen validieren durch Schema-Sprachen: XML Schema Definition (u.a. Erweiterbarkeit, Namensräume, Validierung von Verweisstrukturen), JSON Schema
- Zugriff auf XML-Dokumente mit XPath
- Transformation von XML mit XSLT (u.a. Template-Regeln, Template-Prozessor, Template-Design-Strategien, Modell-Transformation, View-Transformation, Client- und Server-seitige Transformation)
- JSON als Transport-View

Leistungsnachweis

Klausur (90 min)

# Datenschutz

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Datenschutz
Modulverantwortliche	Ronald Petrlc
Dozent(en)	Ronald Petrlc
Vorkenntnisse	Grundkenntnisse der IT-Sicherheit.
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"><li>- Datenschutz-Grundverordnung und weitere relevante Gesetze</li><li>- Überblick über kryptographische Verfahren</li><li>- Privacy by Design</li><li>- Anonymisierung und Pseudonymisierung in der Praxis</li><li>- Sichere Kommunikation in der Praxis: E-Mail und Messenger</li><li>- Datenschutz im Web: Tracking, Social Plugins und co.</li><li>- Anonymität im Internet</li><li>- Anonyme Bezahlverfahren / Bitcoin</li><li>- Datenschutz-Folgenabschätzung</li></ul>
Literatur	<ul style="list-style-type: none"><li>- Petrlc, Sorge: "Datenschutz: Einführung in technischen Datenschutz, Datenschutzrecht und angewandte Kryptographie", Springer-Vieweg, 2017.</li><li>- Wybitul: „EU-Datenschutz-Grundverordnung im Unternehmen: Praxisleitfaden (Kommunikation &amp; Recht)“, 2016.</li></ul>



Leistungsnachweis

Gruppen-Präsentationen zu unterschiedlichen Themen im Rahmen des Kurses.

Zulassungsvoraussetzung

# Einführung in die Funktionale Programmierung

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Einführung in die Funktionale Programmierung
Modulverantwortliche	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"><li>• Typen, Klassen, Parametrische Polymorphie</li><li>• Funktionen, Definition, Komposition</li><li>• Rekursion, Funktionen höherer Ordnung, Kombinatoren</li><li>• Reine Funktionen und Nebeneffekte, Monaden</li><li>• Auswertungsstrategien: strikte und faule Auswertung, unendliche Listen</li></ul>
Literatur	<ul style="list-style-type: none"><li>• Graham Hutton, Programming in Haskell, Cambridge University Press, Cambridge, England, 2. Auflage, 2016.</li><li>• Brian O'Sullivan, John Goerzen, and Don Stewart, Real World Haskell, O'Reilly, Sebastopol, California, U.S.A., 2008.</li><li>• Richard Bird, Pearls of Functional Algorithm Design, Cambridge University Press, Cambridge, England, 2010.</li><li>• Richard Bird, Introduction to Functional Programming using Haskell, Prentice Hall Europe, Harlow, Essex, England, 2. Auflage, 1998.</li><li>• Miran Lipovaca, Learn You a Haskell for Great Good, no starch press, San Francisco, California, U.S.A., 2011.</li></ul>
Leistungsnachweis	Mündliche Prüfung (Dauer: 20 Minuten)
Zulassungsvoraussetzung	

# 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"><li>• Kenntniss der speziellen Eigenschaften und Anforderungen von eingebetteten Systemen</li><li>• Hardware-Architekturen für eingebettete Systeme verstehen und bewerten</li><li>• Einfache, hardwarenahe Programme für Mikrocontroller erstellen können</li><li>• Software-Architekturen für eingebettete Systeme kennen und beurteilen</li><li>• Modellierungsparadigmen für eingebettete Systeme kennen und anwenden</li><li>• Fähigkeit zum Entwurf eingebetteter Systeme</li></ul>
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"><li>• Besondere Eigenschaften, Anwendungsbereiche und Anforderungen von eingebetteten Systemen</li><li>• Hardware-Architekturen, Microcontroller</li><li>• Software für eingebettete Systeme</li><li>• Design / Modellierung eingebetteter Systeme</li><li>• Echtzeitsysteme</li></ul>

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	<ul style="list-style-type: none"><li>• Usability, Ästhetik</li><li>• Interaction Design, User Experience</li><li>• Expressivität von Formen und Farben</li><li>• Wahrnehmungspsychologie/Gestaltpsychologie</li><li>• Designtheorie, Kunsttheorie, Bildtheorie</li><li>• Visuelle Kommunikation</li><li>• Datenvisualisierung</li></ul>
Literatur	<ul style="list-style-type: none"><li>• Sarah Diefenbach, Marc Hassenzahl: Psychologie in der nutzerzentrierten Produktgestaltung</li><li>• Susanne K. Langer: Fühlen und Form. Eine Theorie der Kunst.</li><li>• Rudolf Arnheim: Anschauliches Denken</li><li>• Colin Ware: Information Visualization : Perception for Design</li><li>• Alexander Florin: User - Interface - Design</li></ul>
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"><li>• Fach "Programmieren II"</li><li>• Fach "Web-Programmierung" (oder äquivalente Fähigkeiten in HTML, CSS, JavaScript, PHP)</li></ul> Empfohlen <ul style="list-style-type: none"><li>• Grundlagen SQL (z.B. Fach "Datenbanken")</li></ul>
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"><li>• Fullerton, T. J.: Game Design Workshop, Apple Academic Press Inc., 2013</li><li>• Schell, J.: The Art of Game Design - A Book of Lenses, Morgan Kaufmann, 2014</li><li>• Rabin, S.: Introduction to Game Development, Cengage Delmar, 2009</li><li>• Gregory, J.: Game Engine Architecture, Taylor &amp; Francis Ltd., 2014</li><li>• Nystorm, R.: Game Programming Patterns, Genever Benning, 2014</li></ul>



- 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"><li>• Fachkompetenz: Die Studierenden verstehen die Ursachen von Software-Schwachstellen sowie Konzepte der sicheren Software-Entwicklung und die Grundprinzipien für einen sicheren Entwicklungszyklus.</li><li>• 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.</li><li>• Soziale Kompetenz: Organisation, Planung, Präsentationstechnik.</li></ul>
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"><li>• Software Schwachstellen und Angriffstechniken</li><li>• Verschiedene Arten von Schwachstellen und Angriffen, Kategorisierungen, Common Weakness Enumeration (CWE). Beispiele:<ul style="list-style-type: none"><li>◦ buffer overflows</li><li>◦ integer overflows</li><li>◦ injection attacks</li><li>◦ return-oriented programming</li></ul></li><li>• Sicherheitsaspekte verschiedener Programmiersprachen, z.B.:<ul style="list-style-type: none"><li>◦ Java</li><li>◦ C#</li><li>◦ Rust</li></ul></li><li>• Software Security Testing Tools</li></ul>

- 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. 30 Minuten; 50%)
- schriftliche Ausarbeitung (ca. 10 Seiten; 40%)
- aktive Teilnahme an der Diskussion (10%)

Zulassungsvoraussetzung

# Grundlagen des maschinellen Lernens

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Grundlagen des maschinellen Lernens
Modulverantwortliche	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"><li>• Grundlegende Prinzipien und Theorien des Maschinellen Lernens</li><li>• Einführung der wichtigsten algorithmischen Konzepte</li><li>• Lernen aus Daten</li><li>• Praktische Anwendungen</li></ul>
Literatur	<ul style="list-style-type: none"><li>• Duda, R and Hart, P and Stork, D: Pattern Classification, 2. Auflage, 2001, Wiley, NY</li><li>• Geron A: HAnds-On Machine Learning with Scikit-Learn, Keras &amp; TensorFlow, 2. Auflage, 2019, O'Reilly Media</li><li>• Niemann,H: <i>Klassifikation von Mustern</i>. 2. Überarbeitete Auflage, 2003</li><li>• Goodfellow, I and Bengio,Y and Courville, A: <i>Deep Learning</i>. 2016</li></ul>
Leistungsnachweis	Schriftliche Prüfung (90min) am Veranstaltungsende
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	Vorlesung mit Übungen
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"><li>• Motivation des Softwaretests</li><li>• Grundbegriffe</li><li>• Fundamentaler Testprozess</li><li>• Teststufen und Testarten</li><li>• Statische Qualitätssicherung</li><li>• Testbarkeit von Requirements</li><li>• Testmethoden (Black Box, White Box, erfahrungsbasiert)</li><li>• Konzeption und Planung von Tests</li><li>• Wirtschaftlichkeit des Testens</li><li>• Testrollen</li><li>• Kategorien von Testwerkzeugen.</li></ul>

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 (20 min)

## Grundlagen des Softwaretests (BLOCK)

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Grundlagen des Softwaretests (BLOCK)
Modulverantwortliche	Christian Brandes
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	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.  Themen sind insbesondere: <ul style="list-style-type: none"><li>• Motivation des Softwaretests</li><li>• Grundbegriffe</li><li>• Fundamentaler Testprozess</li><li>• Teststufen und Testarten</li><li>• Statische Qualitätssicherung</li><li>• Testbarkeit von Requirements</li><li>• Testmethoden (Black Box, White Box, erfahrungsbasiert)</li><li>• Konzeption und Planung von Tests</li><li>• Wirtschaftlichkeit des Testens</li><li>• Testrollen</li><li>• Kategorien von Testwerkzeugen.</li></ul>
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, Bucsecs: Basiswissen Testautomatisierung. dpunkt Verlag

Leistungsnachweis

Mündliche Prüfung im Anschluss an die Blockveranstaltung (20 min)



# 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
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 die Nutzung von In-Memory Systemen, insbesondere an praktischen Übungen mit SAP HANA. Zum einen werden analytische Anwendungen durch Modellierung (ohne</p>

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 als Stored Procedures mit der Programmiersprache SQLScript programmiert.

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  
P. Silvia: Einführung in SAP HANA - die neue Einführung. 3. Auflage. Galileo Press, Bonn, 2017  
T. Schneider, E. Westenberg, H. Gahm: ABAP Entwicklung für SAP HANA. Galileo Press, Bonn, 2013

#### Leistungsnachweis

Mündliche Prüfung (Gruppenprüfung, bei zwei Personen 30 Minuten)

#### Zulassungsvoraussetzung

# Knowledge Graphen

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Knowledge Graphen
Modulverantwortliche	Thomas Fuhr
Dozent(en)	Thomas Fuhr
Vorkenntnisse	Grundlegende Kenntnisse der math. Logik (z.B. aus dem Modul "Mathematik I") sowie bzgl. Relationaler Datenbanken (z.B. aus dem Modul "Datenbanken")
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenzzeit, 45 Stunden Vor- und Nachbereitung des Lehrstoffs, 40 Stunden Ausarbeitung Mündl./Schriftlicher Beiträge und Prüfungsvorbereitung
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Seminaristischer Unterricht inkl. Vorlesung, theoretischen und praktischen Übungen sowie Ergebnispräsentationen der Teilnehmenden.
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs in Deutsch und/oder Englisch (bitte nachfragen)
Beitrag zu den Zielen des Studiums	Die erworbenen Kenntnisse/Fähigkeiten des Spezialgebiets können in technische, medieninformatische und wirtschaftsbezogene Anwendungen eingebracht werden, z.B. in den folgenden Bereichen: Systemnahe Software, Systemmanagement, Mensch-Maschine-Kommunikation, Medienanalyse und -verstehen, Informationsmanagement.
Lernziel	Kenntnis wesentlicher Schritte und Methoden zum Aufbau und Erhalt von Knowledge Graphen (KG). Kenntnis grundlegender Begriffe und Ansätze der Wissensrepräsentation. Kenntnis des RDF-Graphmodells. Fähigkeit zur Entwicklung von Daten- und Schemagraphen auf Basis von RDF/S und diese in der Sprache Turtle zu formulieren. Kenntnis der Ausdrucksmöglichkeiten der Abfragesprache SPARQL und Fähigkeit hiermit gezielt KG-Inhalte abzufragen. Kennenlernen wichtiger RDF-Vokabulare. Kenntnis der Linked Open Data Principles. Fähigkeit Inhalte relationaler Datenbanken als RDF-Graph zu nutzen. Kenntnis ausgewählter KG-Anwendungen.
Schlüsselqualifikation	Abstraktes Denken, Umgang mit englischsprachiger Fachliteratur, Fähigkeit eigenes Wissen (ggf. auch auf Englisch) weiter zu vermitteln, Zusammenarbeit im Team.
Lehrinhalte	<i>Knowledge Graphen (KG)</i> sind Strukturen bzw. Systeme, welche große Mengen vernetzter und semantisch beschriebener Daten zusammenfassend repräsentieren und verfügbar machen. KG basieren auf der Idee der Wissensrepräsentation mittels Semantischer Netze, welche bereits in den 1960er Jahren im Gebiet der Symbolischen KI untersucht wurden. Seit Google 2012 seinen <a href="#">Knowledge Graph</a> (zur Unterstützung seiner Informationssuche) vorstellte, hat sich dieser Begriff allgemein für die oben skizzierten Systeme etabliert. Ihre breite Anwendbarkeit (z.B. Informationsintegration und Datenmanagement in Unternehmen, Text- und Sprachanalyse, Informationssuche,

Personal Assistants) wird zunehmend erkannt.

Ein Fundament zur Realisierung von KG ist z.B. der vom [W3C](#) standardisierte Semantic Web-Technologie-Stack, welcher auf dem *Resource Description Framework (RDF)* basiert. Die Vision des *Semantic Web* ist bedeutet, das vorhandene "Web of Documents" um ein "Web of Data" anzureichern (->), "an extension of the current web in which information is given well-defined meaning" (->), d.h. einen über das WWW verteilten Knowledge Graphen aufzubauen.

Diese Lehrveranstaltung behandelt einführend die Fragen:

- Was sind KG? (Grundlagen der Wissensrepräsentation, Semantische Netze, Ontologien, Graphmodelle, aktuelle KG-Definitionen)
- Wie können KG realisiert werden? (Methoden, Technologien, Tools)
- Welche Anwendungen gibt es? (ausgewählte Beispiele)

Anhand des Semantic Web-Technologie-Stacks und der Nutzung einer frei verfügbaren RDF-Datenbank wird konkret und praktisch erarbeitet, wie Daten und Wissen

- repräsentiert (RDF 1.1 Modell & Schema, SKOS, OWL 2),
- ausgetauscht (Serialisierungsformate: RDF/XML, Turtle, u.a.),
- abgefragt (SPARQL 1.1 Query Language),
- vernetzt (RDF Vokabulare, Linked Data, schema.org) und
- veröffentlicht (SPARQL Endpoints, Einbettung in Webseiten),
- aus Relationalen Datenbanken integriert (R2RML) sowie
- durch Automatisches Schlussfolgern ergänzt (RDF Schema Reasoning)

*Labeled-Property-Graphdatenbanken* werden als alternative Realisierungsmöglichkeit vorgestellt.

#### Literatur

- D. Fensel et al.: *Knowledge Graphs*. Springer, 2020.
- A. Blumauer, H. Nagy: *The Knowledge Graph Cookbook*. momochrom,2020.
- M. Kejriwal, C. A. Knoblock, P. Szekely: *Knowledge Graphs*.The MIT Press, 2021.
- Liyang Yu: *A Developer's Guide to the Semantic Web*. Springer, 2011.
- J. Domingue, D. Fensel, J. Hendler (Hrsg.): *Handbook of Semantic Web Technologies*. Springer 2011.
- Bob DuCharme: *Learning SPARQL*. O'Reilly, 2nd Ed. 2013.
- Verschiedene Autoren: *aktuelle W3C Recommendations für RDF, RDFS, SPARQL, R2RML u. a.* [www.w3.org](http://www.w3.org).

#### Leistungsnachweis

Klausur (90 min.)

# Kryptographische Algorithmen und Systeme

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Kryptographische Algorithmen und Systeme
Modulverantwortliche	Hans Löhr
Dozent(en)	Hans Löhr
Vorkenntnisse	Kryptografie: Vorlesung "Kryptographie und Informationssicherheit" oder vergleichbare Vorkenntnisse (Verschlüsselung symmetrisch/asymmetrisch, Digitale Signaturen, MACs, RSA, ElGamal, Diffie-Hellman, Blockchiffren, Hashfunktionen, ...);  Mathematik: Algebra-Kenntnisse (modulare Arithmetik; Gruppen, Ringe Körper, etc.)
Leistungspunkte	5
Semesterwochenstunden	4
Unterrichtssprache	Kurs nur in Deutsch
Lernziel	Vertiefte Kryptographie-Kenntnisse; abstraktes Denken und Problemlösen, mathematische Vorgehensweisen; Präsentation fachlich anspruchsvoller Themen
Lehrinhalte	<p>Kryptografie ist eine wichtige Grundlage für moderne IT-Sicherheit. In dieser Lehrveranstaltung behandeln wir kryptografische Algorithmen und deren Sicherheit sowie zu deren Verständnis nötige Grundlagen aus Mathematik und algorithmischer Algebra. Im Gegensatz zu den klassischen, sehr grundlegenden Verfahren wie RSA, die bereits aus der Grundlagenvorlesung "Kryptographie und Informationssicherheit" bekannt sein sollten, werden in dieser Veranstaltung fortgeschrittene Algorithmen betrachtet, die häufig auf komplizierteren mathematischen Grundlagen beruhen. Viele moderne technische Sicherheitslösungen verwenden bspw. Kryptografie mit sogenannten elliptischen Kurven (z.B. WPA3/Wifi, Bluetooth). Außerdem betrachten wir genauer, wie man die "Sicherheit" solcher Verfahren formalisieren kann, um Sicherheitsbeweise (Reduktionsbeweise) zu führen.</p> <p>Beispielhafte Inhalte der Lehrveranstaltung (die konkreten Themen können variieren):</p> <ul style="list-style-type: none"><li>* Rabins Verschlüsselungs- und Signaturschemata</li><li>* Kramer-Shoup-Kryptosystem und beweisbare IND-CCA2-Sicherheit</li><li>* Grundlagen elliptischer Kurven (EC)</li><li>* Berechnungsverfahren für den diskreten Logarithmus (auf EC)</li><li>* Kryptosysteme wie ECDH, ECIES, ECDSA, Curve25519</li><li>* Beispielhafte Anwendungen und Einsatzszenarien</li><li>* Homomorphe Verschlüsselung (z.B. Pailler-Kryptosystem)</li><li>* Quantencomputer und Post-Quantum-Kryptografie (Ausblick)</li></ul> <p>Die Studierenden sollen in dieser Lehrveranstaltung mathematische Methoden und weiterführendes kryptografisches Fachwissen vermittelt bekommen und erarbeiten.</p>
Literatur	Themenspezifische Fachartikel und Konferenzbeiträge;

Kryptografie-Bücher, z.B.:

- Jonathan Katz, Yehuda Lindell, "Introduction to modern Cryptography", 2008.
- Hans Delfs, Helmut Knebl, "Introduction to Cryptography - Principles and Applications", 2015. (Grundlagen)

Leistungsnachweis

Fachvortrag (Präsentation), aktive Diskussionsteilnahme und mündliche Prüfung

# Lean Startup

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Lean Startup
Modulverantwortliche	Peter Rausch
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 im Team erarbeitet und angewendet:</p> <ul style="list-style-type: none"><li>• Lean Innovation und Design Thinking</li><li>• Lean Startup</li><li>• Business Model Canvas</li><li>• Customer Development</li><li>• Zielgerichtete Interviews: Gesprächsführung und -beobachtung</li><li>• Pitches: Ideen erfolgreich präsentieren</li></ul>
Literatur	<ul style="list-style-type: none"><li>• Steve Blank. The Startup Owner's Manual: The Step-By-Step Guide for Building a Great Company</li><li>• Alexander Osterwalder, Yves Pigneur. Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers</li><li>• Eric Ries. The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses</li><li>• Frank Rimalovski, Giff Constable, Tom Fishburne. Talking to Humans: Success starts with understanding your customers</li></ul>
Leistungsnachweis	2 Kurzpräsentationen (jeweils 15 Min.)

Referat (70 Min.)  
Anwesenheitspflicht

Zulassungsvoraussetzung      Keine



## Moderne Versionen der Logik

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Moderne Versionen der Logik
Modulverantwortliche	Yvonne Stry
Vorkenntnisse	Mathematik 1 und 2  (die Belegung des FWPFs „Logik“ ist NICHT Voraussetzung für die Teilnahme am FWPF „Moderne Versionen der Logik“)
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenzzeit, 85 Stunden Vor- und Nachbearbeitung des Lehrstoffs und Vorbereitung einer Präsentation
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Vorlesung, Übungen, Seminar, Moodle-Kurs, Blockseminar, Online-Veranstaltung
Semesterturnus	Winter- bzw. Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Theoretischer Hintergrund
Lernziel	Kenntnis der grundlegenden Richtungen in der Entwicklung der modernen Logik mit Schwerpunkt auf der Modallogik
Schlüsselqualifikation	
Lehrinhalte	<p>Seit etwa 100 Jahren gibt es nicht mehr nur die eine (klassische) Logik, sondern verschiedene Logiken, die die klassische Logik ergänzen bzw. sogar ersetzen sollen.</p> <p>Wir starten mit einer kurzen Einführung in die klassische Aussagenlogik und können die Schwierigkeiten mit intuitiv problematischen Tautologien nachvollziehen. Im Rahmen der klassischen Aussagenlogik setzen wir einen Schwerpunkt auf semantische Tableaux. Mit diesem Beweisverfahren kann die Mögliche-Welten-Semantik von Kripke relativ einfach formuliert werden. Danach werden verschiedene Ansätze zur Erweiterung der klassischen Logik beschrieben.</p> <p>Im Einzelnen:</p> <ul style="list-style-type: none"><li>· Kurze Einführung in die Aussagenlogik</li><li>· Paradoxien der Implikation und Folgerungen daraus</li></ul>

- Elementare Modallogik
- Diverse Modallogiken
- Striktes Konditional
- Intuitionistische Logik
- Dreiwertige Logiken
- evtl. FDE (First Degree Entailment, Implikation 1. Stufe)
- Fuzzy logic/unendlich-wertige Logik

Literatur

Hughes/Cresswell, A New Introduction to Modal Logic, 1996

Priest, An Introduction to Non-Classical Logic, 2008

Leistungsnachweis

Zur Prüfung erhalten die Studierenden individuell oder zu zweit eine Aufgabe (einen Beweis einer Tautologie oder die Konstruktion eines Gegenbeispiels) und müssen zu einer vorgegebenen Fragestellung im Plenum ein Kurzreferat halten.

Zulassungsvoraussetzung

## Netzwerksicherheit

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Netzwerksicherheit
Modulverantwortliche	Hans Löhr
Vorkenntnisse	Voraussetzung sind Kryptographiekennntnisse (z.B. aus der Grundlagenvorlesung "Kryptographie und Informationssicherheit") und Grundkenntnisse zu Rechnernetzen / Kommunikationssystemen.
Arbeitsaufwand	
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Seminaristischer Unterricht: Vorlesung mit Übungen und Studienarbeit bzw. Vortrag
Semesterturnus	Wintersemester
Unterrichtssprache	Kurs in Deutsch und/oder Englisch (bitte nachfragen)
Beitrag zu den Zielen des Studiums	
Lernziel	<ul style="list-style-type: none"><li>•  Fachkompetenz: Die Studierenden verstehen Technologien und Protokolle zur Absicherung von Netzwerken.</li><li>•  Methodenkompetenz: Die Studierenden erlernen die Fähigkeit, Problemstellungen aus dem Bereich Netzwerksicherheit zu analysieren, Sicherheitstechniken, -tools und -protokolle auszuwählen und kritisch zu evaluieren; darüberhinaus üben sie die Fähigkeit, selbständig komplexe Probleme zu bearbeiten.</li></ul>
Schlüsselqualifikation	
Lehrinhalte	Die Studierenden erhalten in dem Kurs einen vertiefenden Einblick in verschiedene Themenfelder aus dem Bereich Netzwerksicherheit. Ein besonderer Schwerpunkt liegt dabei auf anwendungsorientierten kryptographischen Protokollen, für deren Verständnis Kenntnisse gängiger kryptographischer Algorithmen und Verfahren vorausgesetzt werden. Behandelt werden Themen wie beispielsweise: <ul style="list-style-type: none"><li>•  Sicherheitsprotokolle (insbesondere kryptographische Protokolle) auf unterschiedlichen Netzwerkschichten, wie<ul style="list-style-type: none"><li>◦</li></ul></li></ul>

Layer 2: MACsec, Network Access Control, EAP

◦

Layer 3: IPsec, Wireguard

◦

Layer 4: TLS, DTLS

◦

Layers 5-7: SSH, email security

•

Virtual Private Networks (VPN)

•

Border Gateway Protocol (BGP) Security

•

Wireless Security, beispielsweise

◦

WLAN / wifi

◦

Mobile Network Security

◦

Bluetooth Security

•

Authentisierung und Autorisierungsprotokolle

◦

Kerberos

◦

OAuth 2

•

Anonyme Kommunikation (Mix-Netze, TOR)

•

Firewalls

•

Literatur

- 

Grundlagenliteratur

(beinhaltet mehr Themen als in dieser LV behandelt werden):

- 

Claudia Eckert, „IT-Sicherheit – Konzepte, Verfahren, Protokolle“, de Gruyter Oldenbourg, 2018.

- 

Ross Anderson, „Security Engineering - A Guide to Building Dependable Distributed Systems“, 3rd edition, Wiley, 2020.

- 

Paul C. van Oorschot, „Computer Security and the Internet – Tools and Jewels“, Springer, 2020.

- 

Themenspezifische Fachartikel

Leistungsnachweis

30% Studienarbeit/Vortrag und 70% mündliche Prüfung (20 min.)

Zulassungsvoraussetzung

# Programmieren von Mikrocontrollern

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Programmieren von Mikrocontrollern
Modulverantwortliche	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"><li>• 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.</li><li>• 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.</li></ul>
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](http://www.arm.com) ;[www.st.com](http://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
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"><li>• Programmierung betrieblicher Standardsoftware am Beispiel von SAP-Software in der Programmiersprache ABAP<ul style="list-style-type: none"><li>◦ Grundlegende ABAP-Sprachelemente</li><li>◦ Datenbankzugriffe, Data Dictionary</li><li>◦ ABAP Objects</li><li>◦ Web Dynpro für ABAP</li><li>◦ Business Add-ins</li></ul></li><li>• Ausgewählte Aspekte der Technologie von Unternehmenssoftware</li></ul>
Literatur	H. Keller, S. Krüger: ABAP Objects. Galileo Press, 3. Auflage, 2006 <a href="#">R. Weber: Technologie von Unternehmenssoftware. Berlin – Heidelberg (Springer), 2012</a>
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	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.  Die Analysephase nimmt damit eine Schlüsselrolle ein, die durch ihren spezifischen („weichen“) Charakter im Studiengang Informatik besondere Aufmerksamkeit verdient.
Lernziel	Fähigkeit zur Erhebung, Spezifikation und Inspektion von Anforderungen.  Sehr gute Kenntnis der UML und SysML Analysemodelle, ausgewählter Techniken und Methoden des modernen Requirements Engineering. Fertigkeiten im Requirements Management mit dem Werkzeug DOORS Next.  Vorbereitung auf die Prüfung „Certified Requirements Engineer“ ( <a href="https://www.ireb.org/de">https://www.ireb.org/de</a> ) und bei Interesse: anschließende Durchführung der Prüfung (kostenpflichtig!, für Studierende die Hälfte der aktuellen Zertifizierungsgebühr von circa 300,- EUR, Zertifizierer ist derzeit Certible <a href="https://www.certible.com/de/">https://www.certible.com/de/</a> ).
Schlüsselqualifikation	Fähigkeit zur Verbalisierung von Gedanken (gemäß Friedrich Dürrenmatt: „Die Arbeit an der Sprache ist eine Arbeit am Gedanken.“).  Fertigkeit zur Anwendung von UML Modellen in der Analyse sowie von Techniken und Methoden zur Erhebung, Beschreibung und Prüfung von Anforderungen.
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 und SysML Analysemodelle,
- BPMN
- Feature Modellierung
- Requirements Management in DOORS Next
- weitergehende Kenntnisse für das erfolgreiche Absolvieren der Prüfung „Certified Requirements Engineer“, Foundation Level

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, 2021.
- 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%).

# Simulation von Algorithmen des Maschinellen Lernens für autonomes Fahren

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Simulation von Algorithmen des Maschinellen Lernens für autonomes Fahren
Modulverantwortliche	Ramin Tavakoli Kolagari
Dozent(en)	Matthias Bergler, Ramin Tavakoli Kolagari
Vorkenntnisse	<del>Sachkenntnis im Bereich der Verfahrenstechnik für die Entwicklung des Embedded Systems</del>
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenzzeit, 85 Stunden Vor- und Nachbereitung des Lehrstoffs und Prüfungsvorbereitung
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Vorlesung und Übung
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	In dieser Lehrveranstaltung werden die Studierenden gezielt darauf vorbereitet, sich mit den spezifischen Herausforderungen der Entwicklung simulierter autonomer Fahrzeuge. Ein wichtiger Aspekt dieser Lehrveranstaltung liegt darin, den Umgang mit simulierten Sensordaten zu erlernen. Die Studierenden werden mit der Verarbeitung und Auswertung dieser Sensordaten vertraut gemacht, wobei verschiedene maschinelle Lernansätze zum Einsatz kommen. Durch die Verknüpfung unterschiedlicher Domänen des maschinellen Lernens erhalten die Studierenden eine ganzheitliche Perspektive, die ihnen dabei hilft, die Entwicklung simulierter autonomer Fahrzeuge effektiv zu gestalten.
Lernziel	Bei Abschluss des Lernprozesses wird der erfolgreiche Studierende ein erweitertes Verständnis für die Entwicklung simulierter autonomer Fahrzeuge, mit vereinfachtem Funktionsumfang, haben, sowie ein umfangreiches Wissen über die virtuelle Simulationsumgebung Carla erworben haben. Dabei lernen die Studierenden ein zunächst komplexes Problem, aus dem autonomen Fahren, methodisch in verschiedene elementare Teilprobleme zu unterteilen und systematisch zu lösen. Des Weiteren erlangen die Studierenden die Fähigkeiten zur Erstellung von Fahr- und Testszenarien mit steigendem Schwierigkeitsgrad. Hierfür ist eine Analyse der Fahrsituationen mit gängigen Methoden aus der Wissenschaft notwendig. Dadurch erlangen die Studierenden ein grundlegendes Verständnis dafür, was bei der Systementwicklung in Simulationsumgebungen zu berücksichtigende Faktoren sind. Darüber hinaus wird ein Einblick in die gängigen Techniken der Fahrzeugsteuerung gegeben.
Schlüsselqualifikation	Studierende erlangen die Fähigkeit, die Herausforderungen bei der Entwicklung autonomer Fahrzeuge in einer simulierten Umgebung mit simulierten Sensordaten unter vorgegebenen Rahmenbedingungen zu verstehen. Sie sammeln eigene Erfahrungen in der Anwendung relevanter Technologien aus dem Bereich des autonomen Fahrens, bspw. automatische Spurerkennung.
Lehrinhalte	Das Modul bietet den Studierenden einen Einblick in die Entwicklung und Evaluierung

autonom fahrender Fahrzeuge mithilfe der CARLA Simulationsumgebung. CARLA ist eine Open-Source-Simulationsplattform basierend auf der Spiele-Engine Unity, die speziell für die Entwicklung von autonomen Fahrzeugen entwickelt wurde. Sie bietet eine realitätsnahe Umgebung, in der verschiedene Szenarien simuliert und getestet werden können. In der Entwicklungsumgebung wird stufenweise ein autonom fahrendes Fahrzeug entwickelt und in selbst konstruierten Fahrszenarien getestet. Dabei wird der Funktionsumfang des entwickelten Fahrzeugs stetig erweitert, von anfänglichem Geradeausfahren bis hin zum autonomen Reagieren in spezifischen Fahrsituationen, wie beispielsweise dem Ausweichen von Hindernissen auf der Fahrbahn.

Literatur	Carla Dokumentation und Tutorials
Leistungsnachweis	Testszenarien (bestanden, nicht bestanden) Schriftliche Prüfung
Zulassungsvoraussetzung	

# Software-Ergonomie in der Medizintechnik

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Software-Ergonomie in der Medizintechnik
Modulverantwortliche	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"><li>• Usability Engineering, Jacob Nielsen, 1993</li><li>• Don't make me think, Steve Krug, 2006</li><li>• Florian Sarodnick, Methoden der Usability Evaluierung, Huber, 2006</li><li>• Hoa Loranger, Jakob Nielsen: Web Usability, Addison-Wesley, 2008</li><li>• The Usability Engineering Lifecycle, Deborah J. Mayhew, Morgan Kaufmann, 1999</li></ul>
Leistungsnachweis	<ul style="list-style-type: none"><li>• schriftliche Klausur (90 Minuten Dauer)</li><li>• Ergebnisdokumentation der Projektarbeit (Umfang ca. 10 Seiten)</li></ul>

## 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: <a href="https://www.technical-computing.de">https://www.technical-computing.de</a>
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	<a href="https://www.technical-computing.de/references.html">https://www.technical-computing.de/references.html</a>
Leistungsnachweis	Mündliche Prüfung (30 Minuten)
Zulassungsvoraussetzung	

# Testverfahren für komplexe Softwaresysteme

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Testverfahren für komplexe Softwaresysteme
Modulverantwortliche	Matthias Meitner
Dozent(en)	Matthias Meitner
Vorkenntnisse	Programmierkenntnisse, Grundlagen Software Engineering
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenzzeit, 85 Stunden Vor- und Nachbereitung des Lehrstoffs und Anfertigen der Studienarbeit
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Seminaristischer Unterricht: Vorlesung mit Übungen
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Beim Erstellen von komplexen Programmen sind Softwarefehler unvermeidlich, und die Fähigkeit, diese Fehler zu identifizieren und zu beheben, ist von grundlegender Bedeutung für den Erfolg eines Studierenden. Fehlerhafte Software kann schwerwiegende Konsequenzen haben, sei es in der Unternehmenswelt, in sicherheitskritischen Anwendungen oder im Alltag. Durch das Erlernen von Testmethoden und -werkzeugen sind Studierende in der Lage, hochqualitative Software zu entwickeln und so potenzielle Probleme und Sicherheitsrisiken zu minimieren.
Lernziel	Die Studierenden verstehen die grundlegenden Konzepte und Prinzipien des Softwaretests, einschließlich Testarten, Teststufen, Testziele und Teststrategien. Sie sind in der Lage, bestehende Systeme zu analysieren und dazu passende Tests in verschiedenen Phasen des Softwareentwicklungsprozesses zu erstellen, einschließlich Unit-Tests, Integrationstests, Systemtests und Akzeptanztests. Die Studierenden können spezialisierte Tests wie Sicherheits- und Leistungstests durchführen, um Software möglichst robust und sicher zu machen. Sie können verschiedene Werkzeuge in realen Projekten einsetzen, um komplexe Systeme umfassend zu testen.
Schlüsselqualifikation	Fähigkeit zum Verständnis komplexer Systeme, Analyse und Klassifikation von Problemen und Identifikation von Lösungen, Softskills wie kritisches Denken, Kommunikation, Teamfähigkeit und Präsentationskompetenz
Lehrinhalte	Unit Testing (z.B. Junit) Mocking (z.B. Mockito) Web Application Testing (z.B. Selenium) API Testing (z.B. Postman) Static Code Analysis (z.B. SonarQube, SonarLint, SpotBugs) Performance Testing (z.B. Apache JMeter) Penetration Testing (z.B. Kali Linux, Metasploit) Consumer Driven Contract Testing (z.B. Pact)

CI (z.B. Jenkins)  
Fuzz Testing (z.B. ClusterFuzz, american fuzzy lop)  
Mutation Testing (z.B. PIT Mutation Testing)  
Behavior-Driven Development (z.B. Cucumber)

Literatur	Online-Dokumentationen der verschiedenen Testwerkzeuge
Leistungsnachweis	Studienarbeit (50%), 30 Minuten Präsentation (40%), Übungsaufgaben (10%)
Zulassungsvoraussetzung	



## Text Analytics

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Text Analytics
Modulverantwortliche	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"><li>• wichtige Begriffe im Bereich des Natural Language Processing zu definieren</li><li>• überwachte und unüberwachte maschinelle Lernverfahren für die Analyse unstrukturierter, natürlichsprachlicher Texte zu erklären, zu bewerten und einzusetzen</li><li>• Textdaten mit Python systematisch aufzubereiten, zu visualisieren und zu analysieren</li><li>• eigenständig Lösungsansätze für analytische Fragestellungen zu natürlichsprachlichen Texten zu entwickeln</li><li>• Machine-Learning-Prozesse in Python umzusetzen</li></ul>
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 60% Projekt, 40% Befragung.

#### Zulassungsvoraussetzung

# Verteilte Systeme - Grundlagen und Algorithmen

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Verteilte Systeme - Grundlagen und Algorithmen
Modulverantwortliche	Michael Zapf
Dozent(en)	Michael Zapf
Vorkenntnisse	Rechnerkommunikation
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenz 30 Stunden Vor- und Nachbereitung des Lehrstoffs 40 Stunden praktische Übungen 5 Stunden für Prüfung und Prüfungsvorbereitung
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	4 SWS seminaristischer Unterricht mit Vorlesung und Übungen
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	<p>Die Vernetzung von Rechnersystemen ist nur ein erster Schritt, um Ressourcen standortübergreifend nutzbar zu machen. Jenseits des einfachen Nachrichtenaustauschs lassen sich diese Rechner koordinieren, um verteilte Dienste zu realisieren; Operationen werden über das Netz ausgelöst, Ergebnisse zurückgeliefert. Dies kann so weit gehen, dass Anwendungen über mehrere Rechnersysteme verteilt zur Ausführung kommen, wobei die Verteilung vollständig transparent wird. Dies zeigt nicht nur Vorteile in der Performanz, sondern auch in der Ausfallsicherheit und in der effizienten Nutzung der verfügbaren Ressourcen.</p> <p>Das Konzept der Verteilung äußert sich nicht nur im Großen, wenn komplette Computersysteme betrachtet werden, sondern auch im Kleinen, insbesondere im Entwurf paralleler Architekturen, wenn Rechenkerne in großer Zahl zu Hochleistungsrechnern zusammengefügt werden. Die Effekte der weiträumigen verteilten Systeme, betreffend die Konsistenz von Datenspeichern, konkurrente Zugriffe, Verklemmungen und Abstimmungsprozesse, sind in sehr ähnlicher Form in diesen Multiprozessor-Architekturen wiederzufinden.</p>
Lernziel	Analyse typischer Strukturen bei verteilten Systemen; Verstehen der besonderen Herausforderungen im Vergleich zu einfachen lokalen Systemen insbesondere in Bezug auf Synchronisation, Konsistenz, Effekte durch uneinheitliche Nachrichtenlaufzeiten; Verstehen, Anwenden und Bewerten diverser Verfahren zur Synchronisierung, Auswahl, Ausschluss und Terminierungserkennung in verteilten Systemen; Untersuchen des Laufzeitverhaltens von verteilten Systemen anhand von Kommunikationsdiagrammen; Erkennung von Systemverklemmungen; Anwenden von Techniken zu deren Vermeidung insbesondere im verteilten Falle, Bewerten der Robustheit verteilter Anwendungen; Verständnis der Erfordernis und Wirkung von Konsistenzgarantien; Auswahl einer für gegebene Szenarien angemessenen Garantie; Vergleichen der Techniken und typischer Szenarien für Peer-to-Peer-Netze.

Schlüsselqualifikation	Fähigkeit zu komplexem Denken, Analyse und Klassifikation von Problemen. Fähigkeit zur Modellierung von komplexen, verteilten Computersystemen mithilfe typischer Metamodelle und Analyse des Verhaltens eines Systems
Lehrinhalte	<ul style="list-style-type: none"> <li>• Was sind verteilte Systeme?</li> <li>• Multicast-Konsistenz</li> <li>• Logische Uhren</li> <li>• Physikalische Uhren</li> <li>• Wechselseitiger Ausschluss</li> <li>• Auswahlverfahren</li> <li>• Deadlocks</li> <li>• Terminierung verteilter Anwendungen</li> <li>• Fehlertolerante verteilte Systeme</li> <li>• Verteilter Speicher und Konsistenzmodelle</li> <li>• Optimistische Replikation</li> <li>• Peer-to-Peer-Architekturen</li> </ul>
Literatur	<p>Tanenbaum, Andrew S. ; van Steen, Maarten: Verteilte Systeme. 2., Aufl. : Pearson Studium, 2007. - ISBN 3827372933</p> <p>Friedemann Mattern: Verteilte Basisalgorithmen. Springer-Verlag, ISBN 3-540-51835-5, 1989</p>
Leistungsnachweis	Schriftliche Prüfung (Klausur, 90 min) am Veranstaltungsende

# Visualisierung

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Visualisierung
Modulverantwortliche	Matthias Teßmann
Vorkenntnisse	Grundlagen der Informatik, Algorithmen und Datenstrukturen, Computergraphik, Programmierkenntnisse (C++, Python), Mathematik (insb. Lineare Algebra, Differential- & Integralrechnung)
Arbeitsaufwand	150 Stunden, davon: 60 Stunden Kontaktzeit 90 Stunden Vor- und Nachbereitung des Lehrstoffes sowie Übungsaufgaben
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Vorlesung und Seminar
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
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	<p>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.</p> <p>Behandelt werden u.a. folgende Themen:</p> <ul style="list-style-type: none"><li>• Visualisierungsszenarien</li><li>• Gitterstrukturen und Interpolation</li><li>• Verfahren für 2D Skalar- und Vektorfelder</li><li>• Verfahren für 3D Skalar- und Vektorfelder</li><li>• Verfahren für multivariate Daten</li><li>• Volumenvisualisierung mit Isoflächen</li></ul>

- 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%)

Zulassungsvoraussetzung

# Web Application Security

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer Schwerpunkt-FWPF
Modul	Web Application Security
Modulverantwortliche	Peter Trommler, Ralf Reinhardt
Vorkenntnisse	<ul style="list-style-type: none"><li>• Grundkenntnisse der Programmierung (Vorlesung 'Programmieren 1 und 2', 1. und 2. Semester)</li><li>• Grundkenntnisse über Computer-Netzwerke und verteilte Anwendungen (Vorlesung 'Rechnerkommunikation', 4. Semester)</li></ul>
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"><li>• OWASP Top-10</li><li>• Injection Angriffe</li><li>• Validierung von Eingaben</li><li>• Cross-Site Angriffe</li></ul> Gegenmaßnahmen <ul style="list-style-type: none"><li>• Web Application Firewalls (WAF)</li><li>• Automatisierte Erkennung von Sicherheitslücken</li></ul>
Literatur	<ul style="list-style-type: none"><li>• Sverre Huseby, Innocent Code: A Security Wake-Up Call for Web Programmers (auch deutsch d-punkt Verlag)</li><li>• The Open Web Application Security Project: <a href="http://www.owasp.org">http://www.owasp.org</a></li><li>• Web Application Exploits and Defenses: <a href="http://google-gruyere.appspot.com/">http://google-gruyere.appspot.com/</a></li></ul>
Leistungsnachweis	Mündliche Prüfung (Dauer: 20 Minuten)





# Analyse von Data Mining Use Cases aus dem Bereich Marketing und Finanzierung

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer wählbar außerhalb Schwerpunkt
Modul	Analyse von Data Mining Use Cases aus dem Bereich Marketing und Finanzierung
Modulverantwortliche	Rainer Groß
Vorkenntnisse	Grundkenntnisse im Bereich Marketing, Finanzierung und der Statistik
Arbeitsaufwand	150 Stunden, davon: 50 Stunden Präsenzzeit 100 Stunden Vor- und Nachbereitung der Vorlesung, des Vortrags zu einer Fallstudie und Abschlussprüfung
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Seminar mit Vorlesungsanteil
Semesterturnus	Wintersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Das Modul unterstützt das Ziel: ...Einsatz von Prinzipien und Methoden zur effizienten Informationsnutzung in Unternehmen..
Lernziel	Nach der Teilnahme an diesem Modul sind die Studierenden in der Lage <ul style="list-style-type: none"><li>• Use Case getrieben sinnvolle Verfahren des Data Minings auszuwählen,</li><li>• den Data Mining Prozess mit Hilfe eines Data Mining Tools umsetzen und</li><li>• kritisch die Qualität und Güte des eingesetzten Verfahrens zu bewerten.</li></ul>
Schlüsselqualifikation	<ul style="list-style-type: none"><li>• Problemlösen (Analyse und Klassifikation von Problemen)</li><li>• Denkweisen (abstraktes und logisches Denken),</li><li>• Umgang mit wissenschaftlicher Literatur ("digital content curation"),</li><li>• Schreiben (wissenschaftliches Schreiben) und</li><li>• Vortragen (Präsentationsfähigkeiten, Fachvorträge).</li></ul>
Lehrinhalte	<ul style="list-style-type: none"><li>• Vorstellung von ausgewählten Data Mining Modellen im Bereich Klassifikation und Assoziation.</li><li>• Diskussion der Modelle in Bezug auf Anwendungsszenarien und Anforderungen an den Dateninput,</li><li>• Vorstellung und Anwendung des CRISP-DM Standards für typische Data Mining Use Cases im Bereich Finanzierung und Marketing mittels des Tools RapidMiner.</li></ul>
Literatur	<ul style="list-style-type: none"><li>• Kotu, V., Deshpande, B. Predictive Analytics and Data Mining: Concepts and Practice with RapidMiner (Paperback). Verlag: ELSEVIER SCIENCE &amp; TECHNOLOGY, United States (2015).</li><li>• Hofmann, M., Klinkenberg, R. (Hrsg.), RapidMiner: Data Mining Use Cases and Business Analytics Applications (Chapman &amp; Hall/CRC Data Mining and Knowledge Discovery Series). Verlag: Chapman and Hall/CRC (2013).</li><li>• Cleve, J., Lämmel, U. Data Mining. Oldenbourg Wissenschaftsverlag (2014).</li></ul>

Leistungsnachweis

Vortrag zur individuellen Fallstudie (45 min.) und schriftliche Ausarbeitung (20 Seiten)  
zu je 50%.

Zulassungsvoraussetzung

# Digital Government

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer wählbar außerhalb Schwerpunkt
Modul	Digital Government
Modulverantwortliche	Heidi Schuhbauer
Vorkenntnisse	Grundkenntnisse Informationstechnik
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenz sowie 85 Stunden Vor- und Nachbereiten des Lehrstoffes sowie Übungsaufgaben
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Seminar
Semesterturnus	Winter- bzw. Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Methoden zur Stärkung des Abstraktionsvermögens und zur Stärkung der analytischen Fähigkeiten
Lernziel	Kenntnis von Digital Government und des Umfeldes; Fähigkeit zur Beurteilung der Bedeutung und des Umfangs von Anwendungen im E-Government; Kenntnis und Fähigkeit zur Analyse praktischer Maßnahmen des E-Governments anhand von Beispielen; Fähigkeit Konzepte für Digital Government-Lösungen zu erarbeiten.
Schlüsselqualifikation	Fähigkeit zu selbständigem Arbeiten; Strategien des Wissenserwerbs; Analyse und Klassifikation von Problemen; kreatives Problemlösen; Präsentationskompetenz
Lehrinhalte	<ul style="list-style-type: none"><li>• Was bedeutet Digital Government ?</li><li>• Welche Bereiche umfasst Digital Government ?</li><li>• Besondere Rahmenbedingungen und Herausforderungen</li><li>• Fallbeispiele</li></ul>
Literatur	Mehlich, Electronic Government Wirtz, Digital Government
Leistungsnachweis	Seminarleistung (Referat mit Handout + Portfolio, Studienarbeit mit dem Plenum , Video Erstellung; Gewichtung Referat mit Handout 60%, Studienarbeit 30%, Video 10 %)

# Digitale Bauwirtschaft

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer wählbar außerhalb Schwerpunkt
Modul	Digitale Bauwirtschaft
Modulverantwortliche	Peter Rausch
Vorkenntnisse	keine
Arbeitsaufwand	150 Stunden, davon: 65 Stunden Präsenz, 85 Stunden Vor- und Nachbereiten
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Vorlesung und Ideenwettbewerb
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs in Deutsch und/oder Englisch (bitte nachfragen)
Beitrag zu den Zielen des Studiums	Vertiefung und Anwendung der Fachkenntnisse im Bereich der Anwendungssysteme und Schulung des interdisziplinären Denkens. Aufzeigen der Querverbindungen zwischen Wirtschaft und IT.
Lernziel	Am Ende der Lehrveranstaltung sollen die Studierenden über fundierte Kenntnisse zu den im Kurs angesprochenen Themenfeldern verfügen. Die Studierenden sollen relevante Begriffe kennen und die Fähigkeit haben, sicher mit ihnen umgehen können. Ferner sollen die Studierenden befähigt werden, das Erlernte in der Arbeitswelt anwenden zu können. Durch Teamarbeiten soll darüber hinaus die Sozialkompetenz der Studierenden gestärkt werden. Die Studierenden sollen in der Lage sein, Probleme der Prozessabläufe und der Administration in der Bauwirtschaft zu analysieren und im Rahmen einer Gruppenarbeit neue Lösungsansätze zu gestalten. Des Weiteren soll das interdisziplinäre Denken geschult werden.
Schlüsselqualifikation	Befähigung zur selbständigen Analyse und Lösungsgestaltung, Klassifikation von Problemen. Förderung der Sozialkompetenz. Interdisziplinarität.
Lehrinhalte	Die Studierenden erhalten in dem Kurs einen vertiefenden Einblick zu folgenden Themenfeldern: - Einsatz moderner IT in der Bauwirtschaft - Prozessgestaltung in der Bauwirtschaft - Nutzung und Entwicklung von satellitengestützten Maschinenführungssystemen - Analysemöglichkeiten mit Business Intelligence - IoT Zudem wird der Einsatz von 3-D-Druck, Drohnen und Bilderkennung in der Bauwirtschaft überblicksartig dargestellt.
Literatur	<ul style="list-style-type: none"><li>• Neubig, Tobia/ Schötteler, Sebastian/ Zeh, Andreas/ Stumpf, Michael und Rausch, Peter: Improving the Efficiency of the Construction Industry by Means of Combining GNSS Technologies with Sensor Networks. In: PositionIT, Jan/Feb 2018, South Africa.</li><li>• Kress, André/ Stauffer, Jonathan/ Rausch, Peter/ Schreiber, Fritz/ Stumpf, Michael: A GNSS-based approach to demolition and deconstruction. In: PositionIT, August</li></ul>

2016, South Africa.

- Rausch, Peter/ Schreiber, Fritz/ Diegelmann, Michael: GNSS for material flow control and landfill management. In: PositionIT, Mar/Apr 2014, South Africa.
- Rausch, Peter/ Schreiber, Fritz/ Diegelmann, Michael: Closed loop controlling approaches for projects in the earth moving and road construction industry; in: Proceedings of the 2nd International Conference on Machine Control and Guidance, March 9th-11th, 2010, University of Bonn.
- Rausch, Peter/ Schreiber, Fritz/ Diegelmann, Michael: New controlling system for earth moving and road construction, in: PositionIT, Aug/Sept 2010.
- Rausch, Peter/ Schreiber, Fritz/ Diegelmann, Michael: Effiziente Prozessgestaltung im Erd- und Straßenbau durch den Einsatz von satellitengestützten Entscheidungsunterstützungssystemen; in: Wirtschaftsinformatik, 50. Jg., H. 4, 2008.

Leistungsnachweis

schriftliche Ausarbeitung und Präsentation

# Digitalisierung und Nachhaltigkeit

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer wählbar außerhalb Schwerpunkt
Modul	Digitalisierung und Nachhaltigkeit
Modulverantwortliche	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"><li>• Identifikation der Möglichkeiten und Grenzen einer zunehmenden Digitalisierung für eine nachhaltige Entwicklung</li><li>• Erlernen von informationstechnischen Lösungen zur Steigerung der Ressourcen- und Energieeffizienz</li><li>• Kennenlernen von innovativen Anwendungssystemen zur Unterstützung des Menschen bei der Verrichtung von Arbeit</li></ul>
Lernziel	Die Studierenden können:  <ul style="list-style-type: none"><li>• die Begriffe "Digitalisierung" und "Nachhaltigkeit" erklären,</li><li>• eine zunehmende Digitalisierung im Kontext von Nachhaltigkeit kritisch einordnen,</li><li>• zu den gesellschaftlichen Herausforderungen informationstechnische Lösungen benennen,</li><li>• wichtige Anwendungsfelder für eine Digitalisierung und zur Erreichung von Nachhaltigkeit identifizieren, und</li><li>• die Risiken durch eine zunehmende Digitalisierung erläutern.</li></ul>
Schlüsselqualifikation	<ul style="list-style-type: none"><li>• 40% Fachkompetenz</li><li>• 30% Methodenkompetenz</li><li>• 30% Sozialkompetenz</li></ul>
Lehrinhalte	<ol style="list-style-type: none"><li>1. Grundlegende Begrifflichkeiten, Abgrenzung und Historie<ol style="list-style-type: none"><li>1.1 Begriffe von Digitalisierung</li><li>1.2 Entwicklung von Industrie 1.0 zu Industrie 4.0</li><li>1.3 Was ist Nachhaltigkeit?</li><li>1.4 Zusammenhänge Ökonomie – Ökologie – Gesellschaft – Technologie</li><li>1.5 Historie zu Nachhaltigkeit</li><li>1.6 UN Sustainable Development Goals (SDGs) / UN-Entwicklungsziele</li><li>1.7 Digitalisierung und Nachhaltigkeit: nachhaltig-digitale Organisationen und Forderungen</li></ol></li></ol>

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%).



# Finanzmathematik

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer wählbar außerhalb Schwerpunkt
Modul	Finanzmathematik
Modulverantwortliche	Matthias Börger
Vorkenntnisse	Mathematik I und II
Arbeitsaufwand	75 Stunden, davon: 30 Präsenz 45 Stunden Vor- und Nachbereitung
Leistungspunkte	5
Semesterwochenstunden	4
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"><li>• Arithmetische und geometrische Folgen und Reihen</li><li>• Zinsrechnung (einfache Verzinsung, Zinseszins, gemischte Zinsrechnung, unterjährliche Verzinsung, Effektivzins)</li><li>• Rentenrechnung (vorschüssige und nachschüssige Rente, Rentenbarwerte, Rentenendwerte)</li><li>• Tilgungsrechnung (Ratentilgung, Annuitätentilgung)</li><li>• Ausblick</li></ul>
Literatur	Schwenkert/Stry, Finanzmathematik kompakt, Springer, 2. Auflage 2016
Leistungsnachweis	Klausur 90 min

# Informatikethik

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer wählbar außerhalb Schwerpunkt
Modul	Informatikethik
Modulverantwortliche	Thomas Matzner
Vorkenntnisse	Grundkenntnisse im Programmieren und der Nutzung von Datenbanken
Arbeitsaufwand	60 Stunden Kontaktzeit 90 Stunden Vor- und Nachbereitung der Vorlesung, Bearbeitung der Hausaufgaben, Seminararbeit und Vortrag darüber
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Es wechseln sich ab: - Vorlesung zur Vermittlung notwendiger Grundlagen - Diskussion vorgegebener Fragestellungen zur Einübung des ethischen Denkens und Argumentierens - Seminarvorträge zu Fallstudien
Semesterturnus	Wintersemester
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"><li>• die Grundtechniken anzuwenden, mit denen man ethische Fragen klären kann,</li><li>• häufige Fehlurteile, etwa das Schließen von Tatsachen auf Moralurteile oder das Urteilen ohne hinreichende Klärung der Interessen und Werte aller von einer Handlung Betroffenen, zu erkennen und zu vermeiden,</li><li>• in der eigenen Rolle bei der Entwicklung und dem Betrieb von Informatiksystemen zu erkennen, welche Handlungsweisen zur Erzielung ethisch wünschenswerter Resultate angebracht sind,</li><li>• die in anderen Rollen bei der Entwicklung und dem Betrieb von Informatiksystemen tätigen Mitmenschen bei ethischem Handeln zu unterstützen,</li><li>• Nutzer von Informatiksystemen dabei zu unterstützen, Chancen und Risiken von deren Einsatz sachlich abzuwägen und ihr eigenes Handeln an ihren Interessen auszurichten,</li><li>• zum politischen Diskurs über den Einsatz von Informatiksystemen konstruktiv und sachbezogen beizutragen.</li></ul>
Schlüsselqualifikation	Fähigkeit, die vom Einsatz von Informatiksystemen betroffenen Werte und Interessen herauszuarbeiten. Fähigkeit, Interessenkonflikte sachlich darzustellen und einer Lösung im Sinn der Betroffenen zuzuführen.
Lehrinhalte	Ethische Fragen rund um den Einsatz von Informatiksystemen sind seit einigen Jahren im öffentlichen Bewusstsein, etwa: <ul style="list-style-type: none"><li>• Informatiksysteme bewerten unsere Kreditwürdigkeit, Berufsleistung und vieles andere. Sind Ungerechtigkeiten, etwa Diskriminierung, damit zwangsläufig</li></ul>

verbunden oder sind sie vermeidbar? Was müssen alle Beteiligten tun, um unerwünschte Folgen des Einsatzes zu vermeiden?

- An unsere Staatsorgane stellen wir hohe Ansprüche, unsere Sicherheit zu gewährleisten. Gleichzeitig wünschen wir uns von ihnen, uns nicht mit Überwachung zu behelligen. Wie lässt sich ein Ausgleich zwischen beiden Zielen herstellen?
- Der freie Austausch von Meinungen und Informationen im Internet soll gewährleistet werden; Zensur soll nicht stattfinden. Gleichzeitig gibt es Befürchtungen, durch „fake news“ und andere interessen geleiteten Inhalte könnten große Teile der Bevölkerung manipuliert werden. Wie lassen sich Freiheit und Kontrolle der Inhalte sinnvoll austarieren?
- Wie sicher muss ein autonomes Fahrzeug gemacht werden? Muss es absolut unfallfrei fahren oder genügt es, gleich gut oder etwas besser als der Mensch zu sein?

Die Ethik bietet ein Handwerkszeug zur Behandlung solcher Fragen.

- Grundlagen der Ethik
  - Ethik und Moral
  - Tatsachen vs. Moralurteile: Das Gesetz von Hume
  - Pflichtethik
  - Konsequentialistische Ethik
  - Diskursethik
  - Gesinnungs-, Tugendethik
  - Akteurs- vs. Institutionenethik
  - Menschenrechte
  - Das Letztbegründungsproblem
- Moral des Algorithmisierens
  - Tätigkeiten und Rollen bei Entwicklung und Betrieb von Informatiksystemen
  - Einsatzzweck vs. Lösungsweg
  - Beobachtbares Verhalten vs. innerer Ablauf
  - Regelgetriebene vs. datengetriebene Lösungswege
  - Menschliche vs. algorithmische Urteile
  - Begründungspflicht für Ergebnisse
  - Verantwortung für Informatiksysteme
  - Handlungsempfehlungen für Entwicklung und Betrieb
  - Whistleblowing
- Das neue Veröffentlichen
  - Qualitativer Unterschied zum alten Veröffentlichen
  - Freie Meinungsäußerung
  - Wirkung von Meinungsäußerungen
  - Illegale Inhalte
  - Fake News
- Datensammeln
  - privater Akteure
  - öffentlicher Stellen

## Literatur

Kutschera, Franz von: Grundlagen der Ethik. 2. Auflage. De Gruyter 1999. (Womöglich schwierig erhältlich, dann durch das folgende ersetzbar:)

Birnbacher, Dieter: Analytische Einführung in die Ethik. 3. Auflage. De Gruyter 2013.

Fenner, Dagmar: Einführung in die Angewandte Ethik. Francke 2010.

Bischof, Norbert: Moral. Ihre Natur, ihre Dynamik und ihr Schatten. Böhlau 2012.

Kahneman, Daniel: Thinking, Fast and Slow. Penguin 2011 (auch deutsche Übersetzungen verfügbar).

Matzner, Thomas: Informatikethik. 2. Auflage. BoD 2020.

Leistungsnachweis

1. Vortrag in Form eines Kolloquiums (50%)
2. Klausur (50%)

Zulassungsvoraussetzung

# Informationssicherheitsmanagement

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer wählbar außerhalb Schwerpunkt
Modul	Informationssicherheitsmanagement
Modulverantwortliche	Ronald Petrlc
Vorkenntnisse	
Arbeitsaufwand	150 Stunden, davon: 65 Stunden in Präsenz, 85 Stunden zur Bearbeitung der Projektarbeit
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Vorlesung / Seminar
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Beitrag zu den Zielen des Studiums	Das Modul Informationssicherheitsmanagement fördert das Verständnis bei Studierenden sicherheitsrelevante Aspekte in ihren Projekt- oder Produktplanungen zu berücksichtigen sowie auf aktuelle Entwicklungen der Sicherheitslage innerhalb oder außerhalb des Unternehmens zu reagieren.
Lernziel	Die Studierenden erhalten grundlegendes Wissen über gängige Standards im Bereich des Informationssicherheitsmanagements. Sie haben die Kompetenz, die Handlungsanweisungen zur Steuerung von Anforderungen aus der Informationssicherheit in der betrieblichen Praxis abzuleiten sowie erste Erfahrungen diese umzusetzen.
Schlüsselqualifikation	Anforderungen der Informationssicherheit im Projekt- und Produktmanagement verstehen Probleme erkennen und Lösungen vorbereiten  - Entscheidungen aufbereiten und gegenüber hohen Hierarchieebenen vertreten
Lehrinhalte	- Grundlagen im Umgang mit Normanforderungen - Einführung in die ISO 27000er-Normen sowie BSI-Grundschrift - Aufbau und Betrieb eines Informationssicherheits-Managementsystems im PDCA-Zyklus - Prozessorientiertes Informationssicherheitsmanagement: Risikomanagement, Behandlung von Sicherheitsvorfällen, etc. - Exkurs: Integration weiterer Managementsysteme am Beispiel des Datenschutz-Managementsystems (ISO 27701)
Literatur	- Kersten, Klett, Reuter, Schröder: „IT-Sicherheitsmanagement nach der neuen ISO 27001“, Springer-Vieweg, 2. Auflage, 2016. - Lang, Löhr: „IT-Sicherheit – Technologien und Best Practices für die Umsetzung in Unternehmen“, Carl Hanser Verlag, 2022.
Leistungsnachweis	Projektarbeit (75%), Kolloquium (25%)

# Introduction to Bionic Computation in Business

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer wählbar außerhalb Schwerpunkt
Modul	Introduction to Bionic Computation in Business
Modulverantwortliche	Patricia Brockmann
Vorkenntnisse	Englisch, Supply Chain Management, Geschäftsprozessmanagement, Statistik
Arbeitsaufwand	150 Stunden, davon: 60 Stunden Präsenz 90 Stunden Projektarbeit, Vorbereitung der Projektpräsentation und des Projektberichts
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	Seminar
Semesterturnus	Wintersemester
Unterrichtssprache	Kurs nur in Englisch
Beitrag zu den Zielen des Studiums	Anwendungsbezogene und praxisorientierte Kompetenzen zur Verwendung analytischer Informationssysteme zur Optimierung von Geschäftsprozessen in Unternehmen
Lernziel	<ul style="list-style-type: none"><li>• Eine Einführung in Algorithmen, die biologischen Systeme der Natur nachahmen.</li><li>• Die praktische Anwendung von Bionic-Computation-Algorithmen mit Hilfe von analytischen Informationssystemen zur Optimierung von Geschäftsprozessen</li></ul>
Schlüsselqualifikation	<ul style="list-style-type: none"><li>• Analyse und Modellierung von betriebswirtschaftlichen Fallstudien</li><li>• Anwendung von analytischen informationssystemen zur Optimierung von Geschäftsprozessen</li><li>• Projektergebnisse schriftlich und mündlich auf English zu präsentieren</li></ul>
Lehrinhalte	<p>Theoretische Grundlagen:</p> <ul style="list-style-type: none"><li>• Evolutionäre Algorithmen</li><li>• Neuronale Netzwerke</li><li>• Schwarmintelligenz (z.B. Ameisenalgorithmen, Partikelschwarmoptimierung)</li><li>• Sentiment Analyse</li></ul> <p>Praktische Übungen:</p> <ul style="list-style-type: none"><li>• Modellierung und Analyse von betriebswirtschaftlichen Fallstudien, um Probleme zu identifizieren</li><li>• Anwendung von Bionic-Computation-Algorithmen zur Problemlösung der Fallstudien</li></ul>
Literatur	<ul style="list-style-type: none"><li>• De Jong, K., "Evolutionary Computaton: A Unified Approach", MIT Press 2016.</li><li>• Dorigo, M., et al., "ANTS 2020, Proceedings of the 12th International Conference on Swarm Intelligence", Springer 2020.</li><li>• Dorigo, M., Stützle, T., "Ant Colony Optimization", Bradford, 2004.</li><li>• Eiben, A.E, and Smith, J.E. "Introduction to Evolutionary Computing", Springe 2015.</li><li>• Hassanien and Emary, "Swarm Intelligence: Principles, Advances and Applications",</li></ul>

CRC Press, 2016.

- Slowik, A., Ed. "Swarm Intelligence Algorithms, a Tutorial", CRC Press, 2020.
- Yang, X., "Nature-Inspired Computation and Swarm Intelligence", Elsevier Academic Press 2020.

Leistungsnachweis

Portfolioprüfung bestehend aus:

- einer schriftlichen Ausarbeitung (Projektportfolio) Gewichtung 60%
- einem Referat (30 Minuten), Gewichtung 40%.

# Optimierung 1

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer wählbar außerhalb Schwerpunkt
Modul	Optimierung 1
Modulverantwortliche	Jochen Gorski, Alexander Hufnagel
Dozent(en)	Jochen Gorski
Vorkenntnisse	Mathematik 1 und 2
Arbeitsaufwand	150 Stunden, davon 60h Präsenz, 100h Vor- und Nachbearbeitung
Leistungspunkte	5
Semesterwochenstunden	4
Veranstaltungstyp	2 SU, 2 Ü
Semesterturnus	Sommersemester
Unterrichtssprache	Kurs nur in Deutsch
Lernziel	<p>Kenntnis wichtiger Begriffe, Denkweisen und Lösungsmethoden der mathematischen Optimierung</p> <p>Anschauliche Vorstellung von den wichtigsten Problemstellungen und Lösungsstrategien</p> <p>Verständnis typischer und besonders effizienter Algorithmen, ihrer Möglichkeiten und Beschränkungen</p> <p>Fähigkeit, die kennengelernten Lösungsmethoden auf konkrete Problemstellungen anzuwenden</p>
Lehrinhalte	<p>Wichtige Themen der Linearen und der Nichtlinearen Optimierung (siehe unten), Ausblick auf spezielle Anwendungen (z.B. Standortoptimierung in der Ebene).</p> <p>Lineare Optimierung</p> <ul style="list-style-type: none"><li>• Standardform linearer Programme, Transformationsmöglichkeiten</li><li>• Simplexverfahren als Standardmethode zur Lösung linearer Optimierungsprobleme graphische Veranschaulichung, Tableau-Formulierung</li><li>• Sensitivitätsanalyse</li></ul> <p>Nichtlineare Optimierung</p> <ul style="list-style-type: none"><li>• Unrestringierte Optimierungsprobleme</li><li>• Konvexe Optimierungsprobleme</li><li>• Optimierungsprobleme mit einer oder mehreren Gleichheits- oder Ungleichheits-Nebenbedingungen</li><li>• Karush-Kuhn-Tucker-Bedingungen</li><li>• Hinreichende Bedingungen, geränderte Hesse-Matrix</li></ul>
Literatur	<ul style="list-style-type: none"><li>• S. Boyd/ L. Vandenberghe: Convex Optimization</li></ul>



- W. Domschke/ A. Drexl/ R. Klein/ A. Scholl: Einführung in Operations Research, Springer Verlag
- H. Hamacher/ K. Klamroth: Lineare Optimierung und Netzwerkoptimierung, Vieweg
- P. Stingl: Operations Research, Fachbuchverlag Leipzig
- H. A. Taha: Operations Research, Pearson Education Inc.
- R. Schwenkert/ Y. Stry: Operations Research kompakt, Springer Gabler Verlag

Leistungsnachweis

Schriftliche Prüfung über 90 Minuten (Modulprüfung)

# Spieltheorie

Studiengang	Bachelor Informatik 2. Studienabschnitt Fachwissenschaftliche Wahlpflichtfächer wählbar außerhalb Schwerpunkt
Modul	Spieltheorie
Modulverantwortliche	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 bestimmten Gewinn zu realisieren, sondern auf der Aufteilung des gemeinsam erzielten Gewinns unter den (typischerweise mehr als zwei) kooperierenden Spielern.</p>

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 :-)

## Bachelorarbeit

Studiengang	Bachelor Informatik 2. Studienabschnitt Abschlussarbeit
Modul	Bachelorarbeit
Modulverantwortliche	All Professors of Faculty IN / Alle IN-Professoren
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
Modulverantwortliche	All Professors of Faculty IN / Alle IN-Professoren
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 <a href="http://www.in.th-nuernberg.de">www.in.th-nuernberg.de</a> 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