

FACHHOCHSCHULE WEDEL

Modulhandbuch
Bachelor Informatik

B_Inf1.0 (01.10.2004 bis 30.09.2007)

Wedel, den 29. November 2012

Inhaltsverzeichnis

Modulverzeichnis nach Modulkürzel	2
Modulverzeichnis nach Modulbezeichnung	2
1 Erläuterungen zu den Modulbeschreibungen	3
2 Modulbeschreibungen	5
2.1 Grundlagen der Mathematik	5
2.1.1 bis 31.03.2008 (Ha)	5
2.1.2 ab 01.04.2008 (Eh)	9
2.2 Angewandte Mathematik	13
2.3 Statistik	17
2.4 Einführung in die Informatik	19
2.5 Programmierung	23
2.6 Algorithmen und Datenstrukturen in C	25
2.7 Objektorientierte Programmierung	27
2.8 Systemsoftware	31
2.8.1 bis 30.09.2006 (Bek)	31
2.8.2 ab 01.04.2007 (Ahr)	35
2.9 Eingebettete Systeme	39
2.9.1 bis 30.09.2008 (Pk)	39
2.9.2 ab 01.10.2008 (Saw)	42
2.10 Einführung in die Systementwicklung	45
2.10.1 bis 30.09.2007 (Iw)	45
2.10.2 ab 01.10.2007 (UH)	49
2.11 Fortgeschrittene Systementwicklung	53
2.12 Informationssysteme	57
2.13 ERP	61
2.14 Rechnernetze	63
2.15 Einführung in die Informationsverarbeitung	67
2.15.1 bis 30.09.2008 (Pk)	67
2.15.2 ab 01.10.2008 (Saw)	71
2.16 Technik der Informationsverarbeitung	75
2.16.1 bis 30.09.2008 (Pk)	75
2.16.2 ab 01.10.2008 (Saw)	79
2.17 Computergrafik	83
2.18 Betriebswirtschaftslehre	87
2.18.1 bis 31.03.2009 (Bau)	87
2.18.2 ab 01.04.2009 (Gh)	91
2.19 Recht	97
2.20 Projektmanagement	101
2.20.1 bis 30.09.2007 (Rb)	101
2.20.2 ab 01.10.2007 (Stl)	105
2.21 Seminar	109
2.22 Bachelor-Thesis	111

1 Erläuterungen zu den Modulbeschreibungen

Im Folgenden wird jedes Modul in tabellarischer Form beschrieben. Die Struktur der Tabelle entspricht den Vorgaben der ASIIN. Die Reihenfolge der Beschreibungen richtet sich nach den Modulkürzeln.

Vor den Modulbeschreibungen sind zwei Verzeichnisse aufgeführt, die den direkten Zugriff auf einzelne Modulbeschreibungen unterstützen sollen. Ein Verzeichnis listet die Modulbeschreibungen nach Kürzel sortiert auf, das zweite Verzeichnis ist nach Modulbezeichnung alphabetisch sortiert.

Die folgenden Erläuterungen sollen die Interpretation der Angaben in einzelnen Tabellenfeldern erleichtern, indem sie die Annahmen darstellen, die beim Ausfüllen der Felder zugrunde gelegt wurden.

Kürzel:	FH-internes, bezogen auf den Studiengang eindeutiges Kürzel des Moduls
Lehrveranstaltungen:	Lehrveranstaltungen, die im Modul zusammen gefasst sind, mit ihrem FH-internen Kürzel und ihrer Bezeichnung
Semester:	Auflistung der Semester, in denen Veranstaltungen des Moduls stattfinden. In Klammern hinter den Semesterangaben jeweils die Kürzel der dort stattfindenden Lehrveranstaltungen
Dozent(in):	Namen der Dozenten, die Lehrveranstaltungen des Moduls anbieten, werden in alphabetischer Reihenfolge angegeben. Die Angabe "Dozenten" weist auf eine wechselnde Zuständigkeit für die Durchführung von Veranstaltungen hin.
Zuordnung zum Curriculum:	Auflistung aller Studiengänge, in denen das Modul auftritt, zusammen mit der Angabe, ob es sich um eine Pflicht- oder Wahlveranstaltung handelt, und den Semestern, in denen Veranstaltungen des Moduls liegen
Lehrform/SWS:	Die SWS der im Modul zusammen gefassten Lehrveranstaltungen werden nach Lehrform summiert angegeben, die Angaben zur Gruppengröße beziehen sich auf die Zahl der Hörer, die an Veranstaltungen des Moduls teilnehmen. Darunter können auch Hörer aus anderen Studiengängen sein. Wenn die Zahl als Bereich angegeben ist, treten in den Veranstaltungen des Moduls unterschiedliche Hörerzahlen im angegebenen Bereich auf.

Arbeitsaufwand:	Der Gesamtarbeitsaufwand in Stunden ergibt sich aus den ECTS-Punkten multipliziert mit 30 (Stunden). Der Zeitaufwand für das Eigenstudium ergibt sich, wenn vom Gesamtaufwand die Präsenzzeiten abgezogen werden. Diese ergeben sich wiederum aus den Semesterwochenstunden (SWS), die multipliziert mit 45 (Minuten) geteilt durch 60 die Präsenzzeit ergeben.
Kreditpunkte:	Der angegebene Wert gibt die Summe der ECTS-Punkte an, die in allen Lehrveranstaltungen des Moduls erzielt werden können.
Voraussetzungen:	Es werden Module und Lehrveranstaltungen genannt, die eine inhaltliche Grundlage für das jeweilige Modul darstellen. Bei Lehrveranstaltungen ist der Hinweis auf das jeweilige Modul enthalten, in dem die Lehrveranstaltung als Bestandteil auftritt.
Lernziele/Kompetenzen:	Es werden stichwortartig die zentralen Lernziele des Moduls genannt. Die Reihenfolge der Darstellung lehnt sich an der Reihenfolge der Veranstaltungen im Modul an, wie sie im Feld Lehrveranstaltungen angegeben ist. So ist eine grobe Zuordnung zwischen Lernzielen und vermittelnder Veranstaltung innerhalb des Moduls gegeben.
Inhalt:	Gliederungsartige Auflistung der wesentlichen Inhalt des Moduls, nach Lehrveranstaltungen gruppiert.
Studien-/Prüfungsleistungen:	Auflistung aller Formen von Leistungsermittlung, die in den Veranstaltungen des Moduls auftreten.
Medienformen:	Auflistung aller Medienformen, die in Veranstaltungen des Moduls eingesetzt werden.
Literatur:	Nach Veranstaltungen des Moduls gruppierte Auflistung der wesentlichen Quellen, die den Studierenden zur Ergänzung zu den Veranstaltungsinhalten empfohlen werden. Es wird keine vollständige Auflistung aller Quellen gegeben, die als Grundlage für die Veranstaltung dienen.

2 Modulbeschreibungen

2.1 Grundlagen der Mathematik

2.1.1 bis 31.03.2008 (Ha)

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Grundlagen der Mathematik
Kürzel:	v10
Lehrveranstaltungen:	u100 Diskrete Mathematik v100 Analysis 1 v101 Lineare Algebra
Semester:	1 (u100, v100), 2 (v101)
Modulverantwortliche(r):	Iven Pockrand
Dozent(in):	Dirk Harms, Sebastian Iwanowski, Iven Pockrand
Sprache:	deutsch
Zuordnung zum Curriculum:	Technische Informatik (Bachelor): Pflicht, 1. Sem., 2. Sem. Medieninformatik (Bachelor): Pflicht, 1. Sem., 2. Sem. Informatik (Bachelor): Pflicht, 1. Sem., 2. Sem.
Lehrform / SWS:	Vorlesung: 8 SWS, Gruppengröße: 100 - 130 Übung: 6 SWS, Gruppengröße: 20
Arbeitsaufwand:	Präsenzstudium: 128 Stunden, Eigenstudium: 292 Stunden
Kreditpunkte:	14
Voraussetzungen:	
Lernziele / Kompetenzen:	<p>Sichere Beherrschung des Umgangs mit reellen Funktionen und der grundlegenden Methoden des Differenzierens und Integrierens; Erwerb der Fähigkeit, mathematische Regeln korrekt anzuwenden und praxisorientierte Problemstellungen in mathematische Beziehungen umzusetzen.</p> <p>Entwicklung der Fähigkeit, ausgehend von einer abstrakt-axiomatischen mathematischen Minimalbasis in sich logisch schlüssige Folgerungen und Beweise herzuleiten; Gewinnung eines soliden inhaltlichen Überblicks über die wesentlichen Begriffe der diskreten Mathematik; Erkenntnis des Bezugs zu Problemstellungen aus der Informatik anhand praktischer Beispiele.</p> <p>Beherrschung der grundlegenden mathematischen Konzepte der Vektor- und Matrizenrechnung, linearer Gleichungssysteme sowie affiner und linearer Abbildungen; Sicht der Relevanz der mathematischen Formalismen für die Lösung praxisorientierter Fragestellungen durch Problembeispiele aus Bereichen wie Computergrafik oder geometrisches Modellieren.</p>

Inhalt:	<p>Analysis 1</p> <ul style="list-style-type: none">• Zahlentypen• Zahlenfolgen<ul style="list-style-type: none">– Bildungsgesetze– Grenzwerte• Funktionen, Relationen<ul style="list-style-type: none">– Funktionstypen– Umkehrfunktion• Differentialrechnung<ul style="list-style-type: none">– Differentiationsregeln– Anwendungen der Differentialrechnung• Integralrechnung<ul style="list-style-type: none">– Integrationsmethoden– Anwendungen der Integralrechnung• Reihen<ul style="list-style-type: none">– Konvergenzkriterien– Potenzreihen• Funktionen mit zwei Variablen<ul style="list-style-type: none">– Partielle Differentiation– Extremwertaufgaben mit Nebenbedingungen <p>Diskrete Mathematik</p> <ul style="list-style-type: none">• Grundbegriffe der Logik<ul style="list-style-type: none">– Aussagen und Prädikate• Beweismethoden• Mengen<ul style="list-style-type: none">– Mengenalgebra– Relationen, Abbildungen, Funktionen• Zahlen<ul style="list-style-type: none">– natürliche Zahlen– ganze Zahlen• Kombinatorik<ul style="list-style-type: none">– Abzählungen– Partitionen• Graphen<ul style="list-style-type: none">– Gerichtete und ungerichtete Graphen– Bäume– Graphen und Matrizen• Algebraische Strukturen<ul style="list-style-type: none">– Gruppen, Ringe, Körper• Polynome
---------	--

	<p>Lineare Algebra</p> <ul style="list-style-type: none"> • Vektoren <ul style="list-style-type: none"> – Operationen – Vektorräume – Analytische Geometrie im \mathbb{R}^3 • Matrixalgebra • Determinanten • Matrixinversion • Lineare Gleichungssysteme <ul style="list-style-type: none"> – Lösungsverhalten – Gauß Algorithmus – Cramer'sche Regel • Eigenwerte • Affine und lineare Abbildungen
Studien-/Prüfungsleistungen:	Klausur
Medienformen:	Tafel, Beamerpräsentation, Overheadfolien, Handout
Literatur:	<p>Analysis 1</p> <ul style="list-style-type: none"> • W. Preuß, G. Wenisch: Lehr- und Übungsbuch Mathematik, Bd. 1 und 2 (Fachbuchverlag Leipzig, 2000) • A. Fetzner, H. Fränkel: Mathematik, Bd. 1 und 2 (Springer, 2003) • L. Papula: Mathematik für Ingenieure, Bd. 1 (Vieweg, 2003) <p>Diskrete Mathematik</p> <ul style="list-style-type: none"> • M. Brill: Mathematik für Informatiker (Hanser Verlag, 2001) • A. Beutelspacher et al.: Diskrete Mathematik für Einsteiger (Springer Verlag, 2004) • J. Matousek et al.: Diskrete Mathematik, Eine Entdeckungsreise (Springer Verlag, 1998) • C. Meinel et al.: Mathematische Grundlagen der Informatik (Teubner Verlag, 2002) <p>Lineare Algebra</p> <ul style="list-style-type: none"> • L. Papula: Mathematik für Ingenieure und Naturwissenschaftler Bd. 1 und 2 (Vieweg, 2001) • W. Preuß, G. Wenisch: Lehr- und Übungsbuch Mathematik Bd. 3 (Fachbuchverlag Leipzig, 2001) • W. Preuß, G. Wenisch: Lehr- und Übungsbuch Mathematik für Informatiker (Fachbuchverlag Leipzig, 1997) • G. Farin, D. Hansford: Lineare Algebra: Ein geometrischer Zugang (Springer, 2003)

2.1.2 ab 01.04.2008 (Eh)

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Grundlagen der Mathematik
Kürzel:	v10
Lehrveranstaltungen:	u100 Diskrete Mathematik v100 Analysis 1 v101 Lineare Algebra
Semester:	1 (u100, v100), 2 (v101)
Modulverantwortliche(r):	Iven Pockrand
Dozent(in):	Eike Harms, Sebastian Iwanowski, Iven Pockrand
Sprache:	deutsch
Zuordnung zum Curriculum:	Technische Informatik (Bachelor): Pflicht, 1. Sem., 2. Sem. Medieninformatik (Bachelor): Pflicht, 1. Sem., 2. Sem. Informatik (Bachelor): Pflicht, 1. Sem., 2. Sem.
Lehrform / SWS:	Vorlesung: 8 SWS, Gruppengröße: 100 - 130 Übung: 6 SWS, Gruppengröße: 20
Arbeitsaufwand:	Präsenzstudium: 128 Stunden, Eigenstudium: 292 Stunden
Kreditpunkte:	14
Voraussetzungen:	
Lernziele / Kompetenzen:	<p>Sichere Beherrschung des Umgangs mit reellen Funktionen und der grundlegenden Methoden des Differenzierens und Integrierens; Erwerb der Fähigkeit, mathematische Regeln korrekt anzuwenden und praxisorientierte Problemstellungen in mathematische Beziehungen umzusetzen.</p> <p>Entwicklung der Fähigkeit, ausgehend von einer abstrakt-axiomatischen mathematischen Minimalbasis in sich logisch schlüssige Folgerungen und Beweise herzuleiten; Gewinnung eines soliden inhaltlichen Überblicks über die wesentlichen Begriffe der diskreten Mathematik; Erkenntnis des Bezugs zu Problemstellungen aus der Informatik anhand praktischer Beispiele.</p> <p>Beherrschung der grundlegenden mathematischen Konzepte der Vektor- und Matrizenrechnung, linearer Gleichungssysteme sowie affiner und linearer Abbildungen; Sicht der Relevanz der mathematischen Formalismen für die Lösung praxisorientierter Fragestellungen durch Problembeispiele aus Bereichen wie Computergrafik oder geometrisches Modellieren.</p>

Inhalt:	<p>Diskrete Mathematik</p> <ul style="list-style-type: none">• Grundbegriffe der Logik<ul style="list-style-type: none">– Aussagen und Prädikate• Beweismethoden• Mengen<ul style="list-style-type: none">– Mengenalgebra– Relationen, Abbildungen, Funktionen• Zahlen<ul style="list-style-type: none">– natürliche Zahlen– ganze Zahlen• Kombinatorik<ul style="list-style-type: none">– Abzählungen– Partitionen• Graphen<ul style="list-style-type: none">– Gerichtete und ungerichtete Graphen– Bäume– Graphen und Matrizen• Algebraische Strukturen<ul style="list-style-type: none">– Gruppen, Ringe, Körper• Polynome <p>Analysis 1</p> <ul style="list-style-type: none">• Zahlentypen• Folgen<ul style="list-style-type: none">– Bildungsgesetze– Grenzwerte• Funktionen, Relationen<ul style="list-style-type: none">– Funktionstypen– Umkehrfunktion• Differentialrechnung<ul style="list-style-type: none">– Differentiationsregeln– Anwendungen der Differentialrechnung (Kurvendiskussionen und Extremwerte)• Integralrechnung<ul style="list-style-type: none">– Integrationsmethoden– Anwendungen der Integralrechnung (Bestimmte Integrale)• Funktionen mit zwei Variablen<ul style="list-style-type: none">– Partielle Differentiation– Extremwertaufgaben mit Nebenbedingungen
---------	--

	<p>Lineare Algebra</p> <ul style="list-style-type: none"> • Vektoren <ul style="list-style-type: none"> – Operationen – Vektorräume – Analytische Geometrie im \mathbb{R}^3 • Matrixalgebra • Determinanten • Matrixinversion • Lineare Gleichungssysteme <ul style="list-style-type: none"> – Lösungsverhalten – Gauß Algorithmus – Cramer'sche Regel • Eigenwerte • Affine und lineare Abbildungen
Studien-/Prüfungsleistungen:	Klausur
Medienformen:	Tafel, Beamerpräsentation, Overheadfolien, Handout, Online Video-Mitschnitt der Vorlesung „Analysis 1“ zur eigenständigen Nachbereitung oder zur Wiederholung versäumter Vorlesungsinhalte

Literatur:	<p>Diskrete Mathematik</p> <ul style="list-style-type: none">• M. Brill: Mathematik für Informatiker (Hanser Verlag, 2001)• A. Beutelspacher et al.: Diskrete Mathematik für Einsteiger (Springer Verlag, 2004)• J. Matousek et al.: Diskrete Mathematik, Eine Entdeckungsreise (Springer Verlag, 1998)• C. Meinel et al.: Mathematische Grundlagen der Informatik (Teubner Verlag, 2002) <p>Analysis 1</p> <ul style="list-style-type: none">• A. Fetzer, H. Fränkel: Mathematik, Bd. 1 und 2 (Springer, 2003)• Ohse, D.: Mathematik für Wirtschaftswissenschaftler 1, (Vahlen 2004)• L. Papula: Mathematik für Ingenieure, Bd. 1 (Vieweg, 2003)• W. Preuß, G. Wenisch: Lehr- und Übungsbuch Mathematik, Bd. 1 und 2 (Fachbuchverlag Leipzig, 2000) <p>Lineare Algebra</p> <ul style="list-style-type: none">• L. Papula: Mathematik für Ingenieure und Naturwissenschaftler Bd. 1 und 2 (Vieweg, 2001)• W. Preuß, G. Wenisch: Lehr- und Übungsbuch Mathematik Bd. 3 (Fachbuchverlag Leipzig, 2001)• W. Preuß, G. Wenisch: Lehr- und Übungsbuch Mathematik für Informatiker (Fachbuchverlag Leipzig, 1997)• G. Farin, D. Hansford: Lineare Algebra: Ein geometrischer Zugang (Springer, 2003)
------------	---

2.2 Angewandte Mathematik

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Angewandte Mathematik
Kürzel:	v11
Lehrveranstaltungen:	v111 Operations Research 1 v110 Aufgabe OR1 v112 Numerische Mathematik
Semester:	2 (v110, v111), 3 (v112)
Modulverantwortliche(r):	Hans-Detlef Gerhardt
Dozent(in):	René Bodaine, Hans-Detlef Gerhardt
Sprache:	deutsch
Zuordnung zum Curriculum:	Informatik (Bachelor): Pflicht, 2. und 3. Semester
Lehrform / SWS:	Vorlesung: 6 SWS, Gruppengröße: 95 Übung: 2 SWS, Gruppengröße: 20 Aufgabe: 0 SWS, Gruppengröße: 2
Arbeitsaufwand:	Präsenzstudium: 74 Stunden, Eigenstudium: 166 Stunden
Kreditpunkte:	8
Voraussetzungen:	Analysis 1 (in Modul v10)
Lernziele / Kompetenzen:	<p>Kenntnis der mathematischen Methoden des Operations Research; Fähigkeit, Problemstellungen als OR-Aufgaben zu erkennen, mathematische Modelle zu entwickeln und diese so in standardisierte Modelle zu transformieren, dass die gelernten Lösungsverfahren angewandt werden können;</p> <p>Entwicklung der Fähigkeit, im Team komplexe Optimierungsprobleme zu analysieren und dafür die mathematischen Modelle so zu entwickeln und zu transformieren, dass sie unter Verwendung eines Softwaresystems gelöst werden können.</p> <p>Kenntnis der wesentlichen mathematischen Aufgabenklassen, die einer numerischen Lösung zugänglich sind; Verständnis der mathematischen Grundlagen der numerischer Verfahren und deren algorithmische Umsetzung; Entwicklung der Fähigkeit, Algorithmen hinsichtlich ihrer Geschwindigkeit, Genauigkeit und eventuell nur begrenzter Einsetzbarkeit bewerten zu können;</p> <p>Erwerb des Basiswissens, um ein vertiefendes Studium spezieller Verfahren durchführen zu können; den Einsatz spezieller mathematischer Software kennen lernen und für Problemlösungen anwenden können.</p>

Inhalt:	<p>Operations Research 1</p> <ul style="list-style-type: none"> • Einführung in Operations Research • Lineare Optimierung <ul style="list-style-type: none"> – Mathematische Grundlagen – Lösungsverfahren (Simplex-Methode) • Das Transportproblem <ul style="list-style-type: none"> – Transportmodell und Lösungsverfahren – Erweiterung des Transportproblems • Das Zuordnungsproblem <ul style="list-style-type: none"> – Mathematisches Modell und Lösungsverfahren – Erweiterung des Zuordnungsproblems • Netzplantechnik <ul style="list-style-type: none"> – Begriffe und Verfahren der NPT – Struktur- und Zeitplanung mit Vorgangsknotennetzplänen – Kapazitätsplanung und Kostenplanung • Sensitivitätsanalyse • Tourenplanung <ul style="list-style-type: none"> – Saving/P-Verfahren – Zufallsverfahren mit einer 2-OPT-Regel • Zuschnittsoptimierung <ul style="list-style-type: none"> – Allgemeine Problemstellung – Ganzzahlige Optimierung – Lösungsverfahren • Nichtlineare Optimierung <ul style="list-style-type: none"> – Allgemeine Problemstellung – Lösungsverfahren der quadratischen Optimierung - Gradientenverfahren <p>Numerische Mathematik</p> <ul style="list-style-type: none"> • Fehlerabschätzungen • Lösung nichtlineare Gleichungen <ul style="list-style-type: none"> – Banachscher Fixpunktsatz – Newtonsches Verfahren – Sekantenverfahren • Lösung linearer Gleichungssysteme <ul style="list-style-type: none"> – LR-Zerlegung – Pivotstrategien – Cholesky-Zerlegung – Tridiagonale Gleichungssysteme – Iterationsverfahren • Lösung nichtlinearer Gleichungssysteme <ul style="list-style-type: none"> – Banachscher Fixpunktsatz – Newtonsches Verfahren • Interpolation durch kubische Splines • Approximation • Lösung von Anfangswert-Differenzialgleichungssystemen <ul style="list-style-type: none"> – Allgemeine Einschrittverfahren – Runge-Kutta-Verfahren
---------	---

Studien-/Prüfungsleistungen:	Übungsabnahme, Klausur
Medienformen:	Tafel, Overheadfolien, Beamerpräsentation
Literatur:	<p>Operations Research 1</p> <ul style="list-style-type: none"> • Ellinger, T., Beuermann, G., Leisten, R.: Operations Research Eine Einführung, Springer, 2003 • Lutz, M.: Operations Research Verfahren verstehen und anwenden, Fortis Verlag, 1998 • Runzheimer, B.: Operations Research, Lineare Planungsrechnung und Netzplan- technik, Simulation und Warteschlangentheorie, Gabler, 1999 • Zimmermann, W.: Operations Research Quantitative Methoden zur Entschei- dungsvorbereitung, Oldenbourg, 1995 <p>Numerische Mathematik</p> <ul style="list-style-type: none"> • G. Engeln-Müllges / K. Niederdrenk / R. Wodicka: Numerik-Algorithmen, Springer, 2004 • Gramlich, Günter / Werner, Wilhelm: Numerische Mathematik mit Matlab, dpunkt.verlag, 2000 • N. Köckler / H.,R. Schwarz: Numerische Mathematik, B.,G. Teubner, 2004

2.3 Statistik

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Statistik
Kürzel:	v12
Lehrveranstaltungen:	v120 Statistik 1
Semester:	2
Modulverantwortliche(r):	Hans-Detlef Gerhardt
Dozent(in):	Hans-Detlef Gerhardt
Sprache:	deutsch
Zuordnung zum Curriculum:	Informatik (Bachelor): Pflicht, 2. Sem.
Lehrform / SWS:	Vorlesung: 3 SWS, Gruppengröße: 135 Übung: 1 SWS, Gruppengröße: 135
Arbeitsaufwand:	Präsenzstudium: 38 Stunden, Eigenstudium: 82 Stunden
Kreditpunkte:	4
Voraussetzungen:	
Lernziele / Kompetenzen:	<p>Kenntnisse der grundlegenden Aufgabenstellungen der deskriptiven Statistik; Kenntnisse sowohl zur Vorbereitung und Durchführung statistische Untersuchungen als auch zur Auswertung deren Ergebnisse; Erwerb der Fähigkeit, auf der Basis vorgegebenen Datenmaterials empirische Verteilungsfunktionen abzuleiten und die Werte von Lage- und Streuungsparameter zu berechnen;</p> <p>Fähigkeit zur Ermittlung der Stärke eines Zusammenhanges zwischen Merkmalen und zur Berechnung eines mathematischen Zusammenhangs mittels Regressionsanalyse;</p> <p>Erwerb von Grundkenntnissen der Wahrscheinlichkeitsrechnung; Erwerb der Fähigkeit, Intervallwahrscheinlichkeiten unter Verwendung der wichtigsten diskreten und stetigen Dichte- und Verteilungsfunktionen zu berechnen.</p>

Inhalt:	<ul style="list-style-type: none"> • Einführung in die Statistik <ul style="list-style-type: none"> – Was ist Statistik? – Statistische Untersuchungen • Grundlagen der beschreibende Statistik <ul style="list-style-type: none"> – Statistische Einheiten und Grundgesamtheiten – Merkmale – Darstellungsformen • Kennzeichnende Verteilungen und Maße <ul style="list-style-type: none"> – Häufigkeitsverteilungen und Verteilungsfunktionen – Lageparameter und Streuungsparameter von Häufigkeitsverteilungen – Momente • Der rechnerische Zusammenhang von Merkmalen <ul style="list-style-type: none"> – Korrelationsanalyse – Regressionsanalyse • Wahrscheinlichkeitsrechnung <ul style="list-style-type: none"> – Grundbegriffe – Wahrscheinlichkeitssätze und Rechenregeln – Kombinatorik • Theoretische Verteilungen <ul style="list-style-type: none"> – Grundlagen – Diskrete Wahrscheinlichkeitsverteilungen – Stetige Wahrscheinlichkeitsverteilungen
Studien-/Prüfungsleistungen:	Klausur
Medienformen:	Tafel, Overheadfolien, Beamerpräsentation
Literatur:	<ul style="list-style-type: none"> • Bourier, Günther: Beschreibende Statistik Praxisorientierte Einführung, Gabler, 1999 • Bourier, Günther: Wahrscheinlichkeitsrechnung und schließende Statistik, Gabler, 2001 • Burkschat, Marco u. a.: Beschreibende Statistik Grundlegende Methoden, Springer, 2004. • Kobelt, Helmut: Wirtschaftsstatistik für Studium und Praxis, Schäfer Poeschel, 2000 • Mosler, Schmid: Beschreibende Statistik und Wirtschaftsstatistik, Springer, 2003

2.4 Einführung in die Informatik

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Einführung in die Informatik
Kürzel:	v15
Lehrveranstaltungen:	v150 Grundlagen der Theoretischen Informatik v152 Programmiersprachen 1 v151 Übung Programmiersprachen 1 v153 Automaten und Formale Sprachen
Semester:	1 (v150, v151, v152), 2 (v153)
Modulverantwortliche(r):	Sebastian Iwanowski
Dozent(in):	Andreas Häuslein, Sebastian Iwanowski, Rainer Lang
Sprache:	deutsch
Zuordnung zum Curriculum:	Informatik (Bachelor): Pflicht, 1. Sem., 2. Sem. Technische Informatik (Bachelor): Pflicht, 1. Sem., 2. Sem. Medieninformatik (Bachelor): Pflicht, 1. Sem., 2. Sem. Wirtschaftsinformatik (Bachelor): Pflicht, 1. Sem., 2. Sem.
Lehrform / SWS:	Vorlesung: 7 SWS, Gruppengröße: 80 - 145 Übung: 2 SWS, Gruppengröße: 20
Arbeitsaufwand:	Präsenzstudium: 83 Stunden, Eigenstudium: 217 Stunden
Kreditpunkte:	10
Voraussetzungen:	
Lernziele / Kompetenzen:	<p>Beherrschung der Grundlagen und der Formalisierung logischen Denkens, Fähigkeit, Programme formal zu entwerfen und zu verifizieren, Kenntnis verschiedener Sprachkonzepte, Grundlagen der Komplexitätstheorie, erste Begegnungen mit den Zielen des automatischen Beweizens, Anwendung der Prinzipien Resolution und Unifikation auf kleine Beispiele, Kenntnis der Chancen und der Grenzen logischen Programmierens.</p> <p>Kenntnis der grundlegenden Konzepte imperativer Programmiersprachen und ihrer Umsetzung in Pascal; Fähigkeit zur angemessenen Nutzung dieser Konzepte zum Aufbau vollständiger Programme begrenzter Komplexität.</p> <p>Erwerb von Grundkenntnissen über formale Konzepte der Informatik, Kenntnisse über verschiedene Grammatiken und Sprachklassen, Kenntnisse über Eigenschaften abstrakter Automaten, Zusammenhang zwischen Automaten und Sprachen, Grundbegriffe der Programmsyntax und von Compilern.</p>

Inhalt:	<p>Grundlagen der Theoretischen Informatik</p> <ul style="list-style-type: none">• Logik<ul style="list-style-type: none">– Aussagenlogik– Prädikatenlogik• Grundlagen logischer Programmierung<ul style="list-style-type: none">– Resolution– Unifikation• Sprachkonzepte für die Programmierung<ul style="list-style-type: none">– Verifikation elementarer Anweisungstypen– Verifikation mit Hoare-Tripeln– Verifikation mit vollständiger Induktion– Modularisierung– Rekursion– Klassifizierung in imperative, funktionale und logische Programmiersprachen• Einführung in die Komplexitätstheorie<ul style="list-style-type: none">– Arbeiten mit Landau-Symbolen– Prinzip einer Turing-Maschine– Komplexitätsklassen– NP-Vollständigkeit <p>Programmiersprachen 1</p> <ul style="list-style-type: none">• Grundkonzepte der Datenverarbeitung• Entwurf und Darstellung von Algorithmen• Allgemeine Aspekte von Programmiersprachen• Daten in Programmen<ul style="list-style-type: none">– Grundlegende Datentypen– Variablen, Zuweisungen, Konstanten• Grundsätzlicher Aufbau von Programmen• Operatoren und Ausdrücke• Einfache und strukturierte Anweisungen• Weitere Datentypen und ihre Nutzung<ul style="list-style-type: none">– Strings– Arrays– Records• Strukturierung von Programmen<ul style="list-style-type: none">– Prozeduren und Funktionen– Units
---------	---

	<p>Automaten und Formale Sprachen</p> <ul style="list-style-type: none"> • Endliche Automaten <ul style="list-style-type: none"> – mit und ohne Ausgabe – Deterministische und nichtdeterministische Automaten – Reguläre Sprachen – Reguläre Ausdrücke – Äquivalenz und Minimierung endlicher Automaten – Anwendungen endlicher Automaten – Pumping-Lemma für reguläre Sprachen – Nicht-reguläre Sprachen • Formale Sprachen <ul style="list-style-type: none"> – Ersetzungssysteme – Grammatiken – Chomsky-Hierarchie – Kontextfreie Grammatiken – Normalformen – Pumping-Lemma für kontextfreie Sprachen – Abschlusseigenschaften kontextfreier Sprachen – Entscheidbarkeit bei kontextfreien Sprachen – Eindeutigkeit und Mehrdeutigkeit von Grammatiken – Syntaxanalyse von Programmiersprachen – Kellerautomaten – Kontext-sensitive Sprachen – Linear-beschränkte Automaten – Turingmaschinen – Berechenbarkeit und Entscheidbarkeit
Studien-/Prüfungsleistungen:	Klausur, Übungsabnahme
Medienformen:	Beamerpräsentation, Overheadfolien, Tafel, Handout, studentische Arbeit am Rechner

Literatur:	<p>Grundlagen der Theoretischen Informatik</p> <ul style="list-style-type: none"> • Roland Backhouse: Programmkonstruktion und Verifikation, Hanser 1989, ISBN 3-446-15056-0 Englische Neuauflage: Program Construction: Calculating Implementations from Specifications, Wiley 2003, ISBN 0470848820 • Heinz-Peter Gumm / Manfred Sommer: Einführung in die Informatik, Oldenbourg 2004 (6. Auflage), ISBN 3-486-27389-2 • Michael Huth / Mark Ryan: Logic in Computer Science, Cambridge University Press 2004 (2. Auflage), ISBN 052154310X • Uwe Schöning: Logik für Informatiker, Spektrum 2000 (5. Auflage), ISBN 3-8274-1005-3 <p>Programmiersprachen 1</p> <ul style="list-style-type: none"> • Cooper, Doug; Clancy, Michael: PASCAL, Lehrbuch für das strukturierte Programmieren, 6. Auflage, Vieweg Lehrbuch, 2003 • Herschel, Rudolf; Dieterich, Ernst-Wolfgang: Turbo Pascal 7.0, Oldenbourg, 2000 (2. Auflage) • Ottmann, Thomas; Widmayer, Peter; Programmierung mit PASCAL, Teubner, 2004 <p>Automaten und Formale Sprachen</p> <ul style="list-style-type: none"> • Hopcroft, John E.; Motwani, R.; Ullman, J.D.; Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie, Wesley 2002 (2. Auflage) • Vossen, G., Witt, K.-U., Theoretische Informatik, Verlag Vieweg & Sohn, Braunschweig 2002 • Wegner, I., Theoretische Informatik - eine algorithmenorientierte Einführung, Teubner-Verlag, Stuttgart, 1999 • Lang, R., Automaten und formale Sprachen, Vorlesungsskript, FH-Wedel, 2005
------------	---

2.5 Programmierung

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Programmierung
Kürzel:	v16
Lehrveranstaltungen:	v161 Programmiersprachen 2 v160 Übung Programmiersprachen 2 v162 Programmier-Praktikum
Semester:	2 (v161, v160), 3 (v162)
Modulverantwortliche(r):	Andreas Häuslein
Dozent(in):	Andreas Häuslein
Sprache:	deutsch
Zuordnung zum Curriculum:	Informatik (Bachelor): Pflicht, 2. Sem., 3. Sem. Technische Informatik (Bachelor): Pflicht, 2. Sem. Medieninformatik (Bachelor): Pflicht, 2. Sem., 3. Sem. Wirtschaftsinformatik (Bachelor): Pflicht, 2. Sem.
Lehrform / SWS:	Vorlesung: 4 SWS, Gruppengröße: 105 Übung: 2 SWS, Gruppengröße: 20 Programmier-Praktikum: 0 SWS, Gruppengröße: 1
Arbeitsaufwand:	Präsenzstudium: 56 Stunden, Eigenstudium: 184 Stunden
Kreditpunkte:	8
Voraussetzungen:	Grundlagen der Theoretischen Informatik (in Modul v15) Programmiersprachen 1 (in Modul v15) Übungen Programmiersprachen 1 (in Modul v15)
Lernziele / Kompetenzen:	<p>Kenntnis und praktischer Einsatz der fortgeschrittenen Konzepte imperativer Programmiersprachen in Form komplexer und dynamischer Datenstrukturen sowie ihrer Umsetzung in Pascal; Grundkenntnisse in objektorientierter Programmierung;</p> <p>Nutzung einer modernen Entwicklungsumgebung zur komponentenbasierten, ereignisorientierten Software-Entwicklung;</p> <p>Fähigkeit zur eigenständigen Strukturierung und Realisierung von vollständigen Software-Systemen mittleren Umfangs ausgehend von einer verbalen Aufgabenstellung; Fähigkeit zur Software-Entwicklung im kleinen Team;</p> <p>Entwicklung eines Problembewusstseins im Hinblick auf die benutzungsgerechte Software-Gestaltung; Kenntnis der Grundregeln benutzungsgerechter Oberflächengestaltung; Fähigkeit zum angemessenen Einsatz von Interaktionselementen in grafischen Oberflächen bei Aufgabenstellungen mittleren Schwierigkeitsgrades.</p>

Inhalt:	<ul style="list-style-type: none"> • Einführung in die Programmierung mit Delphi/Object Pascal • Strukturierte Datentypen <ul style="list-style-type: none"> – Arrays (Vertiefung) – Mengen-Typen – Record-Typen (Vertiefung) • Aspekte benutzungsgerechter Oberflächengestaltung mit Delphi/Object Pascal • Prozedurale Typen • Dateien <ul style="list-style-type: none"> – Textdateien – Typisierte Dateien • Ausnahmefallbehandlung • Zeiger und dynamische Datenstrukturen (Listen) • Objektorientierte Programmierung in Object Pascal <ul style="list-style-type: none"> – Klassen und Instanzen – Vererbung – Dynamische Bindung
Studien-/Prüfungsleistungen:	Übungsabnahme, Klausur, Programmabnahme
Medienformen:	Overheadfolien, Softwaredemonstration, Tafel, Handout, studentische Arbeit am Rechner
Literatur:	<ul style="list-style-type: none"> • Kaiser, R.: Object Pascal mit Delphi, Springer Verlag, 1997 • Matthäus, Wolf G.: Grundkurs Programmieren mit Delphi, Vieweg, 2004 • Redaktion Toolbox [Ed.]: Delphi mit Methode, Pascal lernen und verstehen, C & L, 2004 • Doberenz, Walter; Kowalski, Thomas: Borland Delphi 7, Grundlagen und Profiwissen, Hanser Fachbuchverlag, 2002 • Wirth, N.: Algorithmen und Datenstrukturen, Pascal-Version, Teubner, 2000

2.6 Algorithmen und Datenstrukturen in C

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Algorithmen und Datenstrukturen in C
Kürzel:	v20
Lehrveranstaltungen:	v201 Algorithmen und Datenstrukturen in C v200 Übungen Algorithmen und Datenstrukturen in C
Semester:	3
Modulverantwortliche(r):	Uwe Schmidt
Dozent(in):	Uwe Schmidt
Sprache:	deutsch
Zuordnung zum Curriculum:	Informatik (Bachelor): Pflicht, 3. Sem. Medieninformatik (Bachelor): Pflicht, 3. Sem. Technische Informatik (Bachelor): Pflicht, 3. Sem. Wirtschaftsinformatik (Bachelor): Pflicht, 3. Sem.
Lehrform / SWS:	Vorlesung: 4 SWS, Gruppengröße: 90 Übung: 2 SWS, Gruppengröße: 20
Arbeitsaufwand:	Präsenzstudium: 56 Stunden, Eigenstudium: 184 Stunden
Kreditpunkte:	8
Voraussetzungen:	Programmiersprachen 1 (in Modul v15) Grundlagen der Theoretischen Informatik (in Modul v15)
Lernziele / Kompetenzen:	Praktisch sicheres Beherrschen der wesentlichen Sprachelementen der Programmiersprache C; Kennenlernen der Abläufe und deren Kosten (Zeit / Speicher) bei der Ausführung von Programmen in höheren Programmiersprachen auf den von Neumann-Rechnern; Sicherer Umgang mit dynamischen Datenstrukturen, Zeigern und dynamischer Speicherverwaltung; Erlangen grundlegender Kenntnisse über Algorithmen für Felder, Matrizen, für Such- und Sortieralgorithmen und für Algorithmen zur Implementierung von Mengen und Verzeichnissen; Praktisches Anwenden der Komplexitätstheorie mit qualitativer Abschätzung der Laufzeit- und Speicherplatzeffizienz der verschiedenen Algorithmen.
Inhalt:	<ul style="list-style-type: none"> • Grundkonzepte der Sprache C <ul style="list-style-type: none"> – Einfache Datentypen – Präprozessor – Anweisungen – Ausdrücke • Strukturierte Datentypen <ul style="list-style-type: none"> – Felder und Zeiger – struct und union • Dynamische Datenstrukturen <ul style="list-style-type: none"> – Verkettete Listen – Bäume • Algorithmen für Matrizen • Such- und Sortieralgorithmen <ul style="list-style-type: none"> – Speicherplatz und Zeitabschätzungen • Funktionen und Funktionszeiger <ul style="list-style-type: none"> – Prozedurorganisation
Studien-/Prüfungsleistungen:	Übungsabnahme, Klausur
Medienformen:	Beamerpräsentation, Tafel, Softwaredemonstration, studentische Arbeit am Rechner

Literatur:	<ul style="list-style-type: none">• Uwe Schmidt: Algorithmen und Datenstrukturen in C, Vorlesungsunterlagen im Web: http://www.fh-wedel.de/~si/vorlesungen/c/c.html• Harbison, Samuel; Steele, Guy L.: C - A Reference Manual, 5th edition, Prentice Hall, New Jersey, 2002, ISBN: 0-13-089592-X• Kernighan, Brian W.; Ritchie, Dennis M.: C Programming Language, Prentice Hall, New Jersey, 1998, ISBN: 0-13-110370-9• Sedgewick, R.: Algorithmen, 2. Auflage, 2002, Addison Wesley, ISBN 3-8273-7032-9• Saake, G.; Sattler, K.-U.: Algorithmen und Datenstrukturen, Eine Einführung mit Java, 2004, dpunkt Verlag,
------------	---

2.7 Objektorientierte Programmierung

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Objektorientierte Programmierung
Kürzel:	v21
Lehrveranstaltungen:	v211 Objektorientierte Programmierung v210 Übung Objektorientierte Programmierung v212 Software-Projekt
Semester:	4 (v210,v211), 5 (v212)
Modulverantwortliche(r):	Uwe Schmidt
Dozent(in):	Uwe Schmidt
Sprache:	deutsch
Zuordnung zum Curriculum:	Informatik (Bachelor): Pflicht 4. Sem. Medieninformatik (Bachelor): Pflicht, 4. Sem. Technische Informatik (Bachelor): Pflicht, 4. Sem. Wirtschaftsinformatik (Bachelor): Pflicht, 4. Sem.
Lehrform / SWS:	Vorlesung: 2 SWS, Gruppengröße: 80 Übung: 2 SWS, Gruppengröße: 20 Software-Projekt: 1 SWS, Gruppengröße: 8 - 16 aufgeteilt in 4 - 8 Zweiergruppen
Arbeitsaufwand:	Präsenzstudium: 47 Stunden, Eigenstudium: 313 Stunden
Kreditpunkte:	12
Voraussetzungen:	Algorithmen und Datenstrukturen in C (in Modul v20) für das Software-Projekt (v212): Übung Algorithmen und Datenstrukturen (in Modul v20) Übung Objektorientierte Programmierung (in Modul v21)
Lernziele / Kompetenzen:	Vorlesung und Übung OOP: Methodisch fundierter praktischer Umgang mit objektorientierten Programmiersprachen am Beispiel von Java; Beherrschen fortgeschrittener objektorientierter Techniken, wie der Entwicklung von Containerklassen und der Einsatz von Entwurfsmustern; Systematische und korrekte Software-Konstruktion unter Beachtung von Vor- und Nachbedingungen und Ausnahmebehandlung; Grundkenntnisse in nebenläufiger Programmierung am Beispiel von Java threads. Software-Projekt: Einsetzen von OO-Techniken in einer größeren Aufgabe; Entwurf von Datenmodellen und Klassen-Strukturen für eine nichttriviale praxisnahe Problemstellung; Selbständige Projektorganisation mit Aufgabenaufteilung, Zeitplanung, Aufwandschätzung; Verbesserung der Teamfähigkeit; Stärkung der Kommunikationsfähigkeit zwischen Arbeitsgruppen; Koordinierung in einem größeren Team und innerhalb einer Minigruppe; Modularisierung von größeren zusammenhängenden Aufgabenstellungen und Entwicklung von Software-Schnittstellen für diese Modularisierung; Selbständiges Einarbeiten in eine vorgegebene Software-Umgebung.

Inhalt:	<p>Objektorientierte Programmierung</p> <ul style="list-style-type: none"> • Sprachelemente von Java <ul style="list-style-type: none"> – Unicode – Namensräume – Anweisungen und Ausdrücke – Klassen, Objekte und Konstruktoren – Felder – Pakete – Geschachtelte Klassen • Objektorientierte Programmierung <ul style="list-style-type: none"> – Abstrakte Datentypen – Generische ADTs – Vererbung und Wiederverwendung – Mehrfachvererbung und Schnittstellen – Dynamisches Binden – Ist-ein Beziehungen – Vererbung oder Benutzung – Dynamische Datenstrukturen und Containerklassen • OOP mit Java <ul style="list-style-type: none"> – Zusicherungen – Ausnahmen – Laufzeit-Typinformation – Datenströme – Thread-Programmierung • Grafische Oberflächen <ul style="list-style-type: none"> – Ereignisgesteuerte Programmierung – Modell View Controller Muster <p>Software-Projekt</p> <ul style="list-style-type: none"> • Themenstellungen aus verschiedenen Bereichen der Informatik unter Einbeziehung der Software-Systeme in den Laboren der Hochschule. • Es werden größere zusammenhängende Aufgabenstellungen bearbeitet. Dabei werden die Aufgaben in Teilprojekte zerlegt mit definierten Schnittstellen. Diese Teilprojekte werden in Minigruppen von zwei Personen bearbeitet. Die Definition der Schnittstellen zwischen den Minigruppen ist Teil der Aufgabenstellung. Dieses zwingt zur Kommunikation und Abstimmung zwischen den einzelnen Teams und zur sauberen Projektplanung. Die Projekt-Planung und Verfolgung wird durch die Verwendung eines Projekt-Tagebuchs (Resultat eines Projekts) unterstützt.
Studien-/Prüfungsleistungen:	Übungsabnahme, Klausur, Programmdemonstration und mündliche Prüfung
Medienformen:	Beamerpräsentation, Softwaredemonstration, Tafel, studentische Arbeit am Rechner

Literatur:	<ul style="list-style-type: none">• Uwe Schmidt: Objektorientierte Programmierung mit Java, Vorlesungsunterlagen im Web: http://www.fh-wedel.de/~si/vorlesungen/java/java.html• Meyer, Bertrand: Objektorientierte Softwareentwicklung, Hanser, München, 1990, ISBN: 3-446-15773-5• Gosling, James; Joy, Bill; Steele, Guy: The Java Language Specification, 2nd Edition, Addison-Wesley, Reading, 2000, ISBN: 0-201-31008-2• Campione, Mary; Walrath, Kathy: The Java Tutorial, Third Edition, Object-Oriented Programming for the Internet, Addison-Wesley, Reading, 2000, ISBN: 0-201-31007-4• Christian Ullenboom: Java ist auch eine Insel, Programmieren für die Java 2-Plattform in der Version 5, 4. Auflage, Galileo Press GmbH, Bonn, 2004, ISBN: 3-89842-526-6• Mittendorf, Stefan; Singer, Reiner: Java, Programmierhandbuch und Referenz für die Java-2-Plattform, Einführung und Kernpakete, dpunkt Verlag, Heidelberg, 1999, ISBN: 3-920993-82-9• Uwe Schmidt: Software-Praktikum, Themenstellungen im Web: http://www.fh-wedel.de/~si/praktika/SoftwarePraktikum/index.html
------------	--

2.8 Systemsoftware

2.8.1 bis 30.09.2006 (Bek)

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Systemsoftware
Kürzel:	v25
Lehrveranstaltungen:	v250 Betriebssysteme v251 Compilerbau v252 Assembler, Übungen Assembler
Semester:	5
Modulverantwortliche(r):	Wolfgang Ülzmann
Dozent(in):	Daniel Bekowies, Uwe Schmidt, Wolfgang Ülzmann
Sprache:	deutsch
Zuordnung zum Curriculum:	Informatik (Bachelor): Pflicht, 5. Sem.
Lehrform / SWS:	Vorlesung: 6 SWS, Gruppengröße: 20
Arbeitsaufwand:	Präsenzstudium: 56 Stunden, Eigenstudium: 184 Stunden
Kreditpunkte:	8
Voraussetzungen:	
Lernziele / Kompetenzen:	<p>Verständnis für Konzepte der Systemsoftware, sowohl im Bereich der Implementierungsstrategien moderner Multitasking-Betriebssysteme als auch bezüglich der Thematik des Übersetzerbaus und der Formalen Sprachen.</p> <p>Erfassung der Detailfunktionalität wesentlicher Systemfunktionen an ausgewählten Beispielen; Erkennung der Optimierungsmöglichkeiten für die Arbeitsabläufe bei modernen Multitasking-Betriebssystemen; angemessene Einschätzung des Systemverhaltens im Rahmen der Softwareentwicklung und -anwendung; Fähigkeit zur Erkennung und Bewertung der Eigenschaften und Unterschiede realer Betriebssysteme;</p> <p>Tiefgreifendes Verständnis insbesondere der ersten Phasen bei der Übersetzerkonstruktion, nämlich der lexikalischen und der Syntaxanalyse; Fähigkeit zur Nutzung dieser Kenntnisse im Web-Bereich mit den vielen unterschiedlichen Auszeichnungssprachen wie HTML, XML, XSL, XSLT, WML; Fähigkeit zum gewinnbringenden Einsatz dieser Techniken bei der Erstellung von effizienten und zuverlässigen Programmen.</p> <p>Vermittlung von Kenntnissen, die für die systemnahe Programmierung benötigt werden; Erlernen maschinennaher Assembler-Programmierung von Mikroprozessoren; Vermittlung von Erfahrungen im Umgang mit Bits und Bytes und Kennenlernen der Hardwarestrukturen eines Prozessors und deren Funktionalität aus Anwendersicht; Erlangung von Kompetenz, Software auf unterster Ebene zu entwickeln und auszutesten.</p>

Inhalt:	<p>Betriebssysteme</p> <ul style="list-style-type: none"> • Rechnerbetriebsarten • Ablaufsteuerung <ul style="list-style-type: none"> – Prozesse und Threads – Prozessumschalter – Parallelität • Prozess-Synchronisation • Deadlock-Problem <ul style="list-style-type: none"> – Erkennung und Beseitigung – Algorithmen zur Vermeidung • Ein-, Ausgabe-Steuerung <ul style="list-style-type: none"> – Gerätestrukturen – Treiber-Konzepte • Ereignisse und Unterbrechungen <ul style="list-style-type: none"> – Verteilung auf mehrere Prozessoren – Interrupt-Handling • Externe Dateiverwaltung <ul style="list-style-type: none"> – Zugriffsmethoden – Hash-Coding • Arbeitsspeicher-Verwaltung <ul style="list-style-type: none"> – Seitentausch – Austauschstrategien – Speicherverschnitt <p>Compilerbau</p> <ul style="list-style-type: none"> • Compiler im Überblick <ul style="list-style-type: none"> – Compilerphasen – Portierung und Bootstrapping – Compiler und Interpretierer • Grundlagen der formalen Sprachen und der Automatentheorie • Lexikalische Analyse <ul style="list-style-type: none"> – Reguläre Ausdrücke – Nichtdeterministische und deterministische endliche Automaten – Scanner und Scanner-Generatoren • Syntaxanalyse <ul style="list-style-type: none"> – Rekursiver Abstieg – LL- und LR- Parser – Parser-Generatoren • Semantische Analyse <ul style="list-style-type: none"> – Typüberprüfung • Codeerzeugung • Virtuelle Maschinen <p>Assembler</p> <ul style="list-style-type: none"> • Einführung in die Maschinensprache • Arbeitsmodi der CPU <ul style="list-style-type: none"> – Real-Mode – Protected-Mode • Adressierungsarten
32	<ul style="list-style-type: none"> • Zahlensysteme und Darstellung in der CPU / ASP • Anweisungen und Befehle <ul style="list-style-type: none"> – Anweisungen für den Macro Assembler – Anweisungen für den Turbo Assembler – 8086 Befehlsatz

Studien-/Prüfungsleistungen:	Klausur, Testat
Medienformen:	Beamerpräsentation, Tafel, Handout, PDF-Dokumente
Literatur:	<p>Betriebssysteme</p> <ul style="list-style-type: none"> • Tanenbaum: Moderne Betriebssysteme, Verlag Pearson Studium, 2002 • Habermann: Entwurf von Betriebssystemen, Springer-Verlag, 1981 • Wettstein: Architektur von Betriebssystemen, Hanser-Verlag, 1987 • Weck: Prinzipien und Realisierung von Betriebssystemen, Teubner Studienbuch, 1985 <p>Compilerbau</p> <ul style="list-style-type: none"> • Uwe Schmidt: Compilerbau, Vorlesungsunterlagen im Web: http://www.fh-wedel.de/~si/vorlesungen/cb/cb.html • Aho, Alfred V.; Sethi, Ravi; Ullman, Jeffrey D.: Compilers, Principles, Techniques and Tools, Addison-Wesley Longman, Amsterdam, 2000 (Repr.) • Appel, Andrew W.; Palsberg, Jens: Modern Compiler Implementation in Java, 2nd edition, Cambridge University Press, 2002, ISBN: 0-521-82060-X • Wirth, Niklaus: Grundlagen und Techniken des Compilerbaus, Oldenbourg, 1995, ISBN: 3-486-24374-8 <p>Assembler</p> <ul style="list-style-type: none"> • Joachim Rohde, Marcus Roming: Assembler, mitp, 2. Auflage, 2006 • Ernst-Wolfgang Dieterich: Assembler, Oldenbourg, 5. Auflage 2005 • Wolfgang Link: Assembler-Programmierung, Franzis, 12. Auflage 2006 • Reiner Backer: Assembler, Rowohlt Taschenbuch Verlag, 2. Auflage 2007

2.8.2 ab 01.04.2007 (Ahr)

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Systemsoftware
Kürzel:	v25
Lehrveranstaltungen:	v250 Betriebssysteme v251 Compilerbau v252 Assembler, Übungen Assembler
Semester:	5
Modulverantwortliche(r):	Wolfgang Ülzmann
Dozent(in):	Dirk Ahrens, Uwe Schmidt, Wolfgang Ülzmann
Sprache:	deutsch
Zuordnung zum Curriculum:	Informatik (Bachelor): Pflicht, 5. Sem.
Lehrform / SWS:	Vorlesung: 6 SWS, Gruppengröße: 20
Arbeitsaufwand:	Präsenzstudium: 56 Stunden, Eigenstudium: 184 Stunden
Kreditpunkte:	8
Voraussetzungen:	
Lernziele / Kompetenzen:	<p>Verständnis für Konzepte der Systemsoftware, sowohl im Bereich der Implementierungsstrategien moderner Multitasking-Betriebssysteme als auch bezüglich der Thematik des Übersetzerbaus und der Formalen Sprachen.</p> <p>Erfassung der Detailfunktionalität wesentlicher Systemfunktionen an ausgewählten Beispielen; Erkennung der Optimierungsmöglichkeiten für die Arbeitsabläufe bei modernen Multitasking-Betriebssystemen; angemessene Einschätzung des Systemverhaltens im Rahmen der Softwareentwicklung und -anwendung; Fähigkeit zur Erkennung und Bewertung der Eigenschaften und Unterschiede realer Betriebssysteme;</p> <p>Tiefgreifendes Verständnis insbesondere der ersten Phasen bei der Übersetzerkonstruktion, nämlich der lexikalischen und der Syntaxanalyse; Fähigkeit zur Nutzung dieser Kenntnisse im Web-Bereich mit den vielen unterschiedlichen Auszeichnungssprachen wie HTML, XML, XSL, XSLT, WML; Fähigkeit zum gewinnbringenden Einsatz dieser Techniken bei der Erstellung von effizienten und zuverlässigen Programmen.</p> <p>Vermittlung von Kenntnissen, die für die systemnahe Programmierung benötigt werden; Erlernen maschinennaher Assembler-Programmierung von Mikroprozessoren; Vermittlung von Erfahrungen im Umgang mit Bits und Bytes und Kennenlernen der Hardwarestrukturen eines Prozessors und deren Funktionalität aus Anwendersicht; Erlangung von Kompetenz, Software auf unterster Ebene zu entwickeln und auszutesten.</p>

Inhalt:	<p>Betriebssysteme</p> <ul style="list-style-type: none"> • Rechnerbetriebsarten • Ablaufsteuerung <ul style="list-style-type: none"> – Prozesse und Threads – Prozessumschalter – Parallelität • Prozess-Synchronisation • Deadlock-Problem <ul style="list-style-type: none"> – Erkennung und Beseitigung – Algorithmen zur Vermeidung • Ein-, Ausgabe-Steuerung <ul style="list-style-type: none"> – Gerätestrukturen – Treiber-Konzepte • Ereignisse und Unterbrechungen <ul style="list-style-type: none"> – Verteilung auf mehrere Prozessoren – Interrupt-Handling • Externe Dateiverwaltung <ul style="list-style-type: none"> – Zugriffsmethoden – Hash-Coding • Arbeitsspeicher-Verwaltung <ul style="list-style-type: none"> – Seitentausch – Austauschstrategien – Speicherverschnitt <p>Compilerbau</p> <ul style="list-style-type: none"> • Compiler im Überblick <ul style="list-style-type: none"> – Compilerphasen – Portierung und Bootstrapping – Compiler und Interpretierer • Grundlagen der formalen Sprachen und der Automatentheorie • Lexikalische Analyse <ul style="list-style-type: none"> – Reguläre Ausdrücke – Nichtdeterministische und deterministische endliche Automaten – Scanner und Scanner-Generatoren • Syntaxanalyse <ul style="list-style-type: none"> – Rekursiver Abstieg – LL- und LR- Parser – Parser-Generatoren • Semantische Analyse <ul style="list-style-type: none"> – Typüberprüfung • Codeerzeugung • Virtuelle Maschinen <p>Assembler</p> <ul style="list-style-type: none"> • Einführung in die Maschinensprache • Zahlensysteme und Darstellung in der CPU/ASP • Arbeitsmodi der CPU <ul style="list-style-type: none"> – Real-Mode – Protected-Mode
36	<ul style="list-style-type: none"> • Anweisungen und Befehle B_Inf1.0 (01.10.2004 bis 30.09.2007) <ul style="list-style-type: none"> – Anweisungen für den Macro Assembler – 8086 Befehlsatz – 80186 - 80386 Befehlssatz • Adressierungsarten

Studien-/Prüfungsleistungen:	Klausur, Testat
Medienformen:	Beamerpräsentation, Tafel, Handout
Literatur:	<p>Betriebssysteme</p> <ul style="list-style-type: none"> • Tanenbaum: Moderne Betriebssysteme, Verlag Pearson Studium, 2002 • Habermann: Entwurf von Betriebssystemen, Springer-Verlag, 1981 • Wettstein: Architektur von Betriebssystemen, Hanser-Verlag, 1987 • Weck: Prinzipien und Realisierung von Betriebssystemen, Teubner Studienbuch, 1985 <p>Compilerbau</p> <ul style="list-style-type: none"> • Uwe Schmidt: Compilerbau, Vorlesungsunterlagen im Web: http://www.fh-wedel.de/~si/vorlesungen/cb/cb.html • Aho, Alfred V.; Sethi, Ravi; Ullman, Jeffrey D.: Compilers, Principles, Techniques and Tools, Addison-Wesley Longman, Amsterdam, 2000 (Repr.) • Appel, Andrew W.; Palsberg, Jens: Modern Compiler Implementation in Java, 2nd edition, Cambridge University Press, 2002, ISBN: 0-521-82060-X • Wirth, Niklaus: Grundlagen und Techniken des Compilerbaus, Oldenbourg, 1995, ISBN: 3-486-24374-8 <p>Assembler</p> <ul style="list-style-type: none"> • HTML-Referenz http://www.fh-wedel.de/~bek/

2.9 Eingebettete Systeme

2.9.1 bis 30.09.2008 (Pk)

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Eingebettete Systeme
Kürzel:	v26
Lehrveranstaltungen:	v261 Projekt Prozess-Programmierung v262 Prozess-Programmierung
Semester:	4
Modulverantwortliche(r):	Peter Pooch-Haffmans
Dozent(in):	Peter Pooch-Haffmans
Sprache:	deutsch
Zuordnung zum Curriculum:	Informatik (Bachelor): Pflicht, 4. Sem. Technische Informatik (Bachelor): Pflicht, 4. Sem.
Lehrform / SWS:	Vorlesung: 4 SWS, Gruppengröße: 30 - 50
Arbeitsaufwand:	Präsenzstudium: 38 Stunden, Eigenstudium: 82 Stunden
Kreditpunkte:	4
Voraussetzungen:	Programmierung (Modul v16)
Lernziele / Kompetenzen:	<p>Vermittlung von Kenntnissen, die für die systemnahe Programmierung benötigt werden; insbesondere diejenigen Softwarekenntnisse, die für die Programmierung von Embedded Systems, Systemen mit kleinen Hardwareressourcen und Echtzeitsystemen erforderlich sind.</p> <p>Erlernen maschinennaher Assembler-Programmierung von Mikroprozessoren und -controllern; Vermittlung von Erfahrungen im Umgang mit Bits und Bytes und Kennen lernen alle Hardwarestrukturen eines Prozessors und deren Funktionalität aus Anwendersicht; Erlangung von Kompetenz, Software auf unterster Ebene zu entwickeln und auszutesten.</p> <p>Einführung in die Methodik der Programmierung paralleler Prozesse, unter besonderer Berücksichtigung von Echtzeitanforderungen, Erwerb der Befähigung, einfache Echtzeitsysteme zu konzipieren und zu realisieren; neben Kenntnis wesentlicher theoretischer Konzepte besonders auch Fähigkeit zu deren praktische Umsetzung, die im Projekt Prozess-Programmierung durchgeführt wird.</p> <p>Durch Entwurf und Realisierung eines Echtzeit-Multitasking-Programms Nachweis der Kompetenz, derartige Systeme aus einem Anforderungskatalog zu entwerfen und praktisch umzusetzen; Training der Arbeitskoordination, Teamfähigkeit und sozialen Kompetenz durch die im Projekt praktizierte Teamarbeit.</p>

Inhalt:	<p>Prozess-Programmierung</p> <ul style="list-style-type: none"> • Theoretischer Teil <ul style="list-style-type: none"> – Modellierung mit Petrinetzen – Das Coroutinenkonzept – Gegenseitiger Ausschluss, Semaphore – Erzeuger-Verbraucher-Systeme – Monitore, Rendezvous-Konzept – Struktur von Multitasking-Systemen • Praktischer Teil, konkrete Umsetzung <ul style="list-style-type: none"> – Systemzustände, Taskerzeugung, Zeitdienste – Einsatz von Semaphoren – Mailbox-Kommunikation – Kommunikation und Synchronisation mit Message-Passing – Interrupt-Systeme <p>Projekt Prozess-Programmierung</p> <ul style="list-style-type: none"> • Eine Aufgabe aus dem Umfeld Echtzeit, Multitasking, Simulation wird gestellt und steht dem Studenten als Anforderungskatalog zur Verfügung <ul style="list-style-type: none"> – Koordinierung der Arbeitsgruppe – Struktureller Programmwurf – Kodierung und Test – Erstellung einer Dokumentation – Abnahme durch den Betreuer
Studien-/Prüfungsleistungen:	Klausur, Übungsabnahme, benotete Abschlussaufgabe
Medienformen:	Overheadfolien, Beamerpräsentation, Handout, PDF-Dokumente
Literatur:	<p>Prozess-Programmierung</p> <ul style="list-style-type: none"> • Handbuch für den verwendeten Echtzeit-Betriebssystemkern RTKernel • Wörn, Brinkschulte: Echtzeitsysteme: Grundlagen, Funktionsweisen, Anwendungen, Springer • Ghassemmi-Tabrizi: Realzeit-Programmierung, Springer • Tanenbaum: Moderne Betriebssysteme, Pearson • Stallings: Betriebssysteme, Funktion und Design, Pearson <p>Projekt Prozess-Programmierung</p> <ul style="list-style-type: none"> • Handbuch für den verwendeten Echtzeit-Betriebssystemkern RTKernel

2.9.2 ab 01.10.2008 (Saw)

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Eingebettete Systeme
Kürzel:	v26
Lehrveranstaltungen:	v261 Projekt Prozess-Programmierung v262 Prozess-Programmierung
Semester:	4
Modulverantwortliche(r):	Sergei Sawitzki
Dozent(in):	Timm Bostelmann, Sergei Sawitzki
Sprache:	deutsch
Zuordnung zum Curriculum:	Informatik (Bachelor): Pflicht, 4. Sem. Technische Informatik (Bachelor): Pflicht, 4. Sem.
Lehrform / SWS:	Vorlesung: 2 SWS, Gruppengröße: 30 - 50 Praktikum: 2 SWS, Gruppengröße: 2
Arbeitsaufwand:	Präsenzstudium: 38 Stunden, Eigenstudium: 82 Stunden
Kreditpunkte:	4
Voraussetzungen:	Programmierung (Modul v16)
Lernziele / Kompetenzen:	<p>Vermittlung von Kenntnissen, die für die systemnahe Programmierung benötigt werden; insbesondere diejenigen Softwarekenntnisse, die für die Programmierung von Embedded Systems, Systemen mit kleinen Hardwareressourcen und Echtzeitsystemen erforderlich sind.</p> <p>Erlernen von Modellierungstechniken und -methoden bei Programmierung von nebenläufigen Prozessen; Modellierung und Anwendung von Prozesskommunikations- und Synchronisationsmechanismen; insbesondere die Fähigkeit, Aufgabenstellungen auf Systeme nebenläufiger (kooperierender und konkurrierender) Prozesse abzubilden und diese softwaretechnisch umzusetzen.</p> <p>Einführung in die Methodik der Programmierung paralleler Prozesse, unter besonderer Berücksichtigung von Echtzeitanforderungen, Erwerb der Befähigung, einfache Echtzeitsysteme zu konzipieren und zu realisieren; neben Kenntnis wesentlicher theoretischer Konzepte besonders auch Fähigkeit zu deren praktische Umsetzung, die im Projekt Prozess-Programmierung durchgeführt wird.</p> <p>Durch Entwurf und Realisierung eines Echtzeit-Multitasking-Programms Nachweis der Kompetenz, derartige Systeme aus einem Anforderungskatalog zu entwerfen und praktisch umzusetzen; Training der Arbeitskoordination, Teamfähigkeit und sozialen Kompetenz durch die im Projekt praktizierte Teamarbeit.</p>

Inhalt:	<p>Prozess-Programmierung</p> <ul style="list-style-type: none"> • Einleitung <ul style="list-style-type: none"> – Lernziele – Organisatorisches – Literatur – Bezeichnungen und Konventionen – Definition und Einordnung – Historische Entwicklung • Prozesse <ul style="list-style-type: none"> – Grundbegriffe – Technische Umsetzung – Aufgaben des Betriebssystems – Probleme und Lösungsansätze • Kommunikationsmechanismen <ul style="list-style-type: none"> – Übersicht – Semaphore – Monitore – Mailbox-Kommunikation – Nachrichtenaustausch – Weitere Mechanismen – Äquivalenzen und Beispiele • Modellierung <ul style="list-style-type: none"> – Einleitung – Flußdiagramme – Petri-Netze – Weitere Modellierungstechniken • Scheduling <ul style="list-style-type: none"> – Einleitung – Strategien – Zeitverwaltung – Beispiele • Verklemmungen <ul style="list-style-type: none"> – Entstehung – Erkennung und Behandlung – Verhinderung und Vermeidung <p>Projekt Prozess-Programmierung</p> <ul style="list-style-type: none"> • Eine Aufgabe aus dem Umfeld Echtzeit, Multitasking, Simulation wird gestellt und steht dem Studenten als Anforderungskatalog zur Verfügung <ul style="list-style-type: none"> – Koordinierung der Arbeitsgruppe – Struktureller Programmentwurf – Kodierung und Test – Erstellung einer Dokumentation – Abnahme durch den Betreuer
Studien-/Prüfungsleistungen:	Mündliche Prüfung, benotete Abschlussaufgabe
Medienformen:	Beamerpräsentation, Tafel, Handout, PDF-Dokumente

Literatur:	<p>Prozess-Programmierung</p> <ul style="list-style-type: none">• TANENBAUM, Andrew: Moderne Betriebssysteme, Carl Hanser Verlag, 1994• WITZAK, Michael: Echtzeits Betriebssysteme, Franzis' Verlag, 2000• BAUMGARTEN, Bernd: Petri-Netze, Wissenschaftsverlag, 1990• LABROSSE, Jean: MicroC/OS-II, CMP Books, 2002• BECK, Michael; BÖHME, Harald; DZIADZKA, Mirko; KUNITZ, Ulrich: Linux Kernel Programmierung. 6. Auflage, Addison Wesley, 2001 <p>Projekt Prozess-Programmierung</p> <ul style="list-style-type: none">• LABROSSE, Jean: MicroC/OS-II, CMP Books, 2002• Praktikumsanleitung mit Kurzreferenz
------------	--

2.10 Einführung in die Systementwicklung

2.10.1 bis 30.09.2007 (Iw)

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Einführung in die Systementwicklung
Kürzel:	v30
Lehrveranstaltungen:	v300 SW-Engineering v301 Unix + Internet, Übungen Unix + Internet
Semester:	1
Modulverantwortliche(r):	Uwe Schmidt
Dozent(in):	Sebastian Iwanowski, Uwe Schmidt
Sprache:	deutsch
Zuordnung zum Curriculum:	Informatik (Bachelor): Pflicht, 1. Sem. Wirtschaftsinformatik (Bachelor): Pflicht, 1. Sem., 3. Sem.
Lehrform / SWS:	Vorlesung: 6 SWS, Gruppengröße: 100 - 120 Übung: 2 SWS, Gruppengröße: 20
Arbeitsaufwand:	Präsenzstudium: 74 Stunden, Eigenstudium: 166 Stunden
Kreditpunkte:	8
Voraussetzungen:	
Lernziele / Kompetenzen:	<p>Vermittlung eines Überblicks über das gesamte Gebiet, sichere Beherrschung aller für den Berufsalltag relevanten Begriffe bei der Softwareentwicklung, Kenntnis der wesentlichen Merkmale der einzelnen Projektentwicklungsphasen, erste Begegnungen mit den grundlegenden Techniken und Werkzeugen von Systemanalyse und Softwareentwurf.</p> <p>Vermittlung grundlegender Kenntnissen über Betriebssysteme, wie Filter, Pipelines, Prozessverwaltung und Dateisystem; praktische Erfahrungen sammeln bei der Erstellung von Software mit Skriptsprachen; Vorteile und Gefahren kennen lernen bei der Softwareentwicklung und dem Prototyping mit Skriptsprachen; Einführung in die Theorie der regulären Ausdrücke und Kennenlernen der Mächtigkeit und der Grenzen von regulären Ausdrücken, Beherrschen von regulären Ausdrücken für die Verarbeitung von Texten und Auszeichnungssprachen.</p> <p>Erlangen grundlegender Kenntnisse über die Architektur und Arbeitsweise von Web-Anwendungen.</p>

Inhalt:	<p>SW-Engineering:</p> <ul style="list-style-type: none">• Grundlegende Prinzipien<ul style="list-style-type: none">– Systeme und Modelle– Abstraktion– Zerlegung und Perspektivenbildung• Softwareplanung<ul style="list-style-type: none">– Lastenheft– Pflichtenheft• Systemanalyse<ul style="list-style-type: none">– Prozessorientierte Modellierungsmethoden: Funktionsbaum, Datenflussdiagramm, Entscheidungstabelle/-baum, Kontrollflussdiagramme– Datenorientierte Modellierungsmethoden: Entity-Relationship-Modellierung, Objektorientierte Modellierung• Systementwurf<ul style="list-style-type: none">– Abgrenzung zur Systemanalyse– Entwurfsrichtungen: top-down und bottom-up– Modularisierung
---------	--

	<p>Unix und Internet:</p> <ul style="list-style-type: none"> • CASE-Tools <ul style="list-style-type: none"> – UML: Klassendiagramme, Use-Case-Diagramme, Sequenzdiagramme, Zustandsübergangsdigramme, Aktivitätsdiagramme – ARIS: Das Aris-Haus mit den wesentlichen Sichten, EPKs, Zusammenspiel mit UML-Bausteinen • Aufwandsabschätzung <ul style="list-style-type: none"> – Basismethoden: Gewichtungsmethode, parametrische Gleichungen, Multiplikatormethode, Analogiemethode, Relationsmethode, Kennzahlenverfahren, Prozentsatzverfahren – Function-Point-Methode • Qualitätsmanagement <ul style="list-style-type: none"> – Verschiedene Qualitätskriterien – Blackbox-Verfahren – Whitebox-Verfahren • Projektmanagement <ul style="list-style-type: none"> – Allgemeine Organisationsmethoden: Wasserfallmodell, Prototyping, Spiralmodell – Spezielle Organisationsmethoden im Vergleich: RUP, XP • Unix <ul style="list-style-type: none"> – Systemstruktur – einfache Shell Kommandos – Dateisystem – Filter und Pipelines – Skriptprogrammierung mit der Shell – Architektur und Arbeitsweise des X-Systems • Reguläre Ausdrücke <ul style="list-style-type: none"> – Reguläre Mengen – Mächtigkeit und Grenzen von regulären Ausdrücken – Suchen, Zerlegen und Editieren mit regulären Ausdrücken • Skriptsprachen <ul style="list-style-type: none"> – Einfache bash-Programme – Einführung in die Skriptsprache Tcl • Software-Werkzeuge <ul style="list-style-type: none"> – make – Versionsverwaltung mit CVS • Web-Anwendungen <ul style="list-style-type: none"> – Einführung in HTML und XML – HTTP – Architektur und Arbeitsweise von Web-Servern – CGI-Programmierung mit Shell und Tcl
Studien-/Prüfungsleistungen:	Klausur, Übungsabnahme
Medienformen:	Beamerpräsentation, Tafel, Softwaredemonstration, studentische Arbeit am Rechner

Literatur:	<p>SW-Engineering</p> <ul style="list-style-type: none">• Helmut Balzert: Lehrbuch der Software-Technik, Band 1: Software-Entwicklung, Spektrum 2000 (2. Auflage), ISBN 3-8274-0480-0• Helmut Balzert: Lehrbuch der Software-Technik, Band 2: Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung, Spektrum 1998, ISBN 3-8274-0065-1• Bernd Brügge / Allen Dutoit: Objektorientierte Softwaretechnik mit UML, Entwurfsmustern und Java, Pearson Studium 2004, ISBN 3-8273-7082-5• Ian Sommerville: Software Engineering Addison-Wesley 2004 (7. Auflage), ISBN 0-321-21026-3 <p>Unix und Internet</p> <ul style="list-style-type: none">• Uwe Schmidt: Unix und Internet, Vorlesungsunterlagen im Web: http://www.fh-wedel.de/~si/Vorlesungen/interne/betriebsinternem• Koller, Michael: Linux, Installation, Konfiguration, Anwendung, 6. überarbeitete und erweiterte Auflage, Addison-Wesley, Bonn, 2002, ISBN: 3-8273-1854-8• Zeller, Andreas; Krinke, Jens: Programmierwerkzeuge, Versionskontrolle, Konstruktion, Testen, Fehlersuche, dpunkt Verlag, Heidelberg, ISBN: 3-932588-70-3• Maurer, Rainer; Paukstadt, Oliver: HTML und CGI-Programmierung, mit CD-ROM, Dynamische WWW-Seiten erstellen mit Tcl, 2. überarbeitete und erweiterte Auflage, dpunkt Verlag, Heidelberg, 1998, ISBN: 3-920993-79-9
------------	--

2.10.2 ab 01.10.2007 (UH)

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Einführung in die Systementwicklung
Kürzel:	v30
Lehrveranstaltungen:	v300 SW-Engineering v301 Unix und Internet, Übungen Unix und Internet v302 Compilerbau
Semester:	1 (v300), 3 (v301)
Modulverantwortliche(r):	Uwe Schmidt
Dozent(in):	Ulrich Hoffmann, Uwe Schmidt
Sprache:	deutsch
Zuordnung zum Curriculum:	Informatik (Bachelor): Pflicht, 1. Sem. Wirtschaftsinformatik (Bachelor): Pflicht, 1. Sem.
Lehrform / SWS:	Vorlesung: 6 SWS, Gruppengröße: 100 - 120 Übung: 2 SWS, Gruppengröße: 20
Arbeitsaufwand:	Präsenzstudium: 74 Stunden, Eigenstudium: 166 Stunden
Kreditpunkte:	8
Voraussetzungen:	
Lernziele / Kompetenzen:	<p>Überblick über das gesamte Gebiet der Softwaretechnik; Beherrschung aller für den Berufsalltag relevanten Begriffe der Softwareentwicklung; Kenntnisse der wesentlichen Merkmale der einzelnen Projektentwicklungsphasen; Kenntnisse agiler Softwareentwicklungsmethoden; Begegnungen mit den grundlegenden Techniken und Werkzeugen des Softwareentwurfs.</p> <p>Vermittlung grundlegender Kenntnissen über Betriebssysteme, wie Filter, Pipelines, Prozessverwaltung und Dateisystem; praktische Erfahrungen sammeln bei der Erstellung von Software mit Skriptsprachen; Vorteile und Gefahren kennen lernen bei der Softwareentwicklung und dem Prototyping mit Skriptsprachen; Einführung in die Theorie der regulären Ausdrücke und Kennenlernen der Mächtigkeit und der Grenzen von regulären Ausdrücken, Beherrschen von regulären Ausdrücken für die Verarbeitung von Texten und Auszeichnungssprachen.</p> <p>Erlangen grundlegender Kenntnisse über die Architektur und Arbeitsweise von Web-Anwendungen.</p> <p>Tiefgreifendes Verständnis insbesondere der ersten Phasen bei der Übersetzerkonstruktion, nämlich der lexikalischen und der Syntaxanalyse; Fähigkeit zur Nutzung dieser Kenntnisse im Web-Bereich mit den vielen unterschiedlichen Auszeichnungssprachen wie HTML, XML, XSL, XSLT, WML; Fähigkeit zum gewinnbringenden Einsatz dieser Techniken bei der Erstellung von effizienten und zuverlässigen Programmen.</p>

Inhalt:	<p>SW-Engineering</p> <ul style="list-style-type: none">• Grundlegende Prinzipien<ul style="list-style-type: none">– Software, Eigenschaften, Qualitätskriterien– Entwicklungsphasen, Systeme und Modelle– Abstraktion– Zerlegung und Perspektivenbildung• Vorgehensmodelle<ul style="list-style-type: none">– Phasenmodelle, Wasserfallmodell– evolutionäres Modell, Prototyping– Spiralmodell, V-Modell– Rational Unified Process– Agile Methoden: eXtreme Programming, Scrum– Modellgetriebene Softwareentwicklung (MDA)• Softwareplanung<ul style="list-style-type: none">– Lastenheft– Pflichtenheft– Aufwandschätzungen, verschiedene Basis-Methoden– Function-Point-Methode• Systemanalyse<ul style="list-style-type: none">– Funktions/Prozessorientierte Modellierungsmethoden: Funktionsbaum, Datenflussdiagramm, Entscheidungstabelle/-baum, Kontrollflussdiagramme– Datenorientierte Modellierungsmethoden: Entity-Relationship-Modellierung, Objektorientierte Modellierung (UML-Klassendiagramme)– Dynamikorientierte Modellierungsmethoden: Petri-Netze, Sequenzdiagramme, Zustandsautomaten (UML)• Systementwurf<ul style="list-style-type: none">– Modularisierung, Kopplung, Kohärenz• Implementierung<ul style="list-style-type: none">– Unit-Tests– Refactoring– Testgetriebene Softwareentwicklung
---------	---

	<p>Unix und Internet</p> <ul style="list-style-type: none"> • Unix <ul style="list-style-type: none"> – Systemstruktur – einfache Shell Kommandos – Dateisystem – Filter und Pipelines – Skriptprogrammierung mit der Shell – Architektur und Arbeitsweise des X-Systems • Reguläre Ausdrücke <ul style="list-style-type: none"> – Reguläre Mengen – Mächtigkeit und Grenzen von regulären Ausdrücken – Suchen, Zerlegen und Editieren mit regulären Ausdrücken • Skriptsprachen <ul style="list-style-type: none"> – Einfache bash-Programme – Einführung in die Skriptsprache Tcl • Software-Werkzeuge <ul style="list-style-type: none"> – make – Versionsverwaltung mit CVS • Web-Anwendungen <ul style="list-style-type: none"> – Einführung in HTML und XML – HTTP – Architektur und Arbeitsweise von Web-Servern – CGI-Programmierung mit Shell und Tcl <p>Compilerbau</p> <ul style="list-style-type: none"> • Compiler im Überblick <ul style="list-style-type: none"> – Compilerphasen – Portierung und Bootstrapping – Compiler und Interpretierer • Grundlagen der formalen Sprachen und der Automatentheorie • Lexikalische Analyse <ul style="list-style-type: none"> – Reguläre Ausdrücke – Nichtdeterministische und deterministische endliche Automaten – Scanner und Scanner-Generatoren • Syntaxanalyse <ul style="list-style-type: none"> – Rekursiver Abstieg – LL- und LR- Parser – Parser-Generatoren • Semantische Analyse <ul style="list-style-type: none"> – Typüberprüfung • Codeerzeugung • Virtuelle Maschinen
Studien-/Prüfungsleistungen:	Klausur, Übungsabnahme
Medienformen:	Beamerpräsentation, Tafel, Softwaredemonstration, studentische Arbeit am Rechner

Literatur:	<p>SW-Engineering</p> <ul style="list-style-type: none"> • BALZERT, Helmut: Lehrbuch der Softwaretechnik, Band 1: Softwareentwicklung, Spektrum 2000 (2. Auflage), ISBN 3-8274-0480-0 • BALZERT, Helmut: Lehrbuch der Softwaretechnik, Band 2: Softwaremanagement, Softwarequalitätssicherung, Unternehmensmodellierung, Spektrum 1998, ISBN 3-8274-0065-1 • BRÜGGE, Bernd; DUTOIT, Allen: Objektorientierte Softwaretechnik mit UML, Entwurfsmustern und Java, Pearson Studium 2004, ISBN 3-8273-7082-5 • SOMMERVILLE, Ian: Software Engineering Addison-Wesley 2004 (7. Auflage), ISBN 0-321-21026-3 • PRESSMAN, Roger S.: Software Engineering - A Practitioner's Approach, McGraw-Hill Companies, Inc. 1997, ISBN 0-07-052182-4 <p>Unix und Internet</p> <ul style="list-style-type: none"> • Uwe Schmidt: Unix und Internet, Vorlesungsunterlagen im Web: http://www.fh-wedel.de/~sie/Vorlesungen/interne/betriebsinternem • Koller, Michael: Linux, Installation, Konfiguration, Anwendung, 6. überarbeitete und erweiterte Auflage, Addison-Wesley, Bonn, 2002, ISBN: 3-8273-1854-8 • Zeller, Andreas; Krinke, Jens: Programmierwerkzeuge, Versionskontrolle, Konstruktion, Testen, Fehlersuche, dpunkt Verlag, Heidelberg, ISBN: 3-932588-70-3 • Maurer, Rainer; Paukstadt, Oliver: HTML und CGI-Programmierung, mit CD-ROM, Dynamische WWW-Seiten erstellen mit Tcl, 2. überarbeitete und erweiterte Auflage, dpunkt Verlag, Heidelberg, 1998, ISBN: 3-920993-79-9 <p>Compilerbau</p> <ul style="list-style-type: none"> • Uwe Schmidt: Compilerbau, Vorlesungsunterlagen im Web: http://www.fh-wedel.de/~si/vorlesungen/cb/cb.html • Aho, Alfred V.; Sethi, Ravi; Ullman, Jeffrey D.: Compilers, Principles, Techniques and Tools, Addison-Wesley Longman, Amsterdam, 2000 (Repr.) • Appel, Andrew W.; Palsberg, Jens: Modern Compiler Implementation in Java, 2nd edition, Cambridge University Press, 2002, ISBN: 0-521-82060-X • Wirth, Niklaus: Grundlagen und Techniken des Compilerbaus, Oldenbourg, 1995, ISBN: 3-486-24374-8
------------	--

2.11 Fortgeschrittene Systementwicklung

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Fortgeschrittene Systementwicklung
Kürzel:	v31
Lehrveranstaltungen:	v310 SW-Design v311 Systemanalyse v312 SW-Technik für Internet-Anwendungen
Semester:	4 (v310, v312), 5 (v311)
Modulverantwortliche(r):	Andreas Häuslein
Dozent(in):	Andreas Häuslein, Uwe Schmidt
Sprache:	deutsch
Zuordnung zum Curriculum:	Informatik (Bachelor): Pflicht, 4. Sem., 5. Sem.
Lehrform / SWS:	Vorlesung: 8 SWS, Gruppengröße: 75 - 85
Arbeitsaufwand:	Präsenzstudium: 74 Stunden, Eigenstudium: 166 Stunden
Kreditpunkte:	8
Voraussetzungen:	Einführung in die Systementwicklung (Modul v30)
Lernziele / Kompetenzen:	<p>Kenntnis der wesentlichen Entwicklungstätigkeiten, die der Implementierung von Software vorgelagert sind; Kenntnis der methodischen Grundlagen dieser Tätigkeiten; Fähigkeit, durch Einsatz des methodischen Instrumentariums, die Grundlage für die Entwicklung qualitativ hochwertiger Software zu legen.</p> <p>Erkennen der Bedeutung und praktischer Umgang mit der Modellbildung im Softwaredesign; Kennenlernen der zentralen Entwurfsmuster und ihrer Anwendungsfelder; Modellierung überschaubarer aber nicht trivialer Fallstudien sowohl mit informellen Methoden, wie UML, als auch mit formalen Spezifikationsmethoden und mit abstrakter Syntax; Entwicklung lauffähiger Prototypen mit der als ausführbarer Spezifikationsprache eingesetzten funktionalen Sprache Haskell.</p> <p>Einschätzungsvermögen hinsichtlich der Notwendigkeit und Grenzen von Systemanalysen, insbesondere in Bezug auf die Gestaltung betrieblicher Informationssysteme; Kenntnis der wesentlichen Techniken zur Informationsgewinnung in Unternehmen einschließlich ihrer Vor- und Nachteile; Kenntnisse der im Unternehmensumfeld praktisch relevanten methodischen Ansätze zur Systemmodellierung und der damit verbundenen Modellnotationen; Fähigkeit zur Nutzung der Modellierungsmittel zum Aufbau von Analysemodellen für wirtschaftliche Problemstellungen begrenzter Komplexität.</p> <p>Kenntnis der technischen Randbedingungen des Internet und ihrer Auswirkungen auf die Entwicklung von Software; Konzept und Vorzüge von Stylesheets kennen; wichtige Konzepte, Sprachen und Architekturen zur Realisierung dynamischer Webseiten kennen und problembezogen auswählen können; Konzepte und XML und damit verbundene anwendungsneutrale Techniken kennen, insbesondere ihre Einsatzmöglichkeiten bewerten können.</p>

Inhalt:	<p>SW-Design</p> <ul style="list-style-type: none"> • Methoden, Techniken und Werkzeuge im Software-Entwurf <ul style="list-style-type: none"> – OMT, UML – formale Methoden – Abstrakte Syntax zur Datenmodellierung • Entwurfsmuster <ul style="list-style-type: none"> – Strukturmuster – Verhaltensmuster – Erzeugungsmuster • Fallstudien (Beispiele) <ul style="list-style-type: none"> – XML Strukturbaum – Indexierer für Freitextsuche – Projekttagbuch – OMT CASE-Tool für Klassendiagramme <p>Systemanalyse</p> <ul style="list-style-type: none"> • Grundbegriffe der Systemanalyse <ul style="list-style-type: none"> – Gegenstand und Zielsetzung im Unternehmensumfeld – Methodische Grundlagen • Systemaufnahme <ul style="list-style-type: none"> – Informationsgewinnung – Untersuchungsbereiche zu Analyse betrieblicher Informationssysteme • Systemmodellierung <ul style="list-style-type: none"> – Ereignisgesteuerte Prozessketten zur Modellierung von Geschäftsprozessen – Strukturierte Analyse und Essenzielle Modellierung – Besonderheiten der Ist-Analyse • Objektorientierte Analyse • Übergang zum Systementwurf <p>SW-Technik für Internetanwendungen</p> <ul style="list-style-type: none"> • Organisation des Internet • Technische Basiskonzepte des WWW <ul style="list-style-type: none"> – Ressourcenidentifikation – HTML – Style Sheets • Dynamik in Web-Seiten <ul style="list-style-type: none"> – Client-seitige Dynamik – Server-seitige Dynamik • XML und damit verbundene Sprachkonzepte <ul style="list-style-type: none"> – Möglichkeiten der Syntaxfestlegung – Verarbeitung von XML-Dokumenten – Transformation von XML-Dokumenten • Sicherheit und Grundlagen der Kryptografie <ul style="list-style-type: none"> – Sicherheitsgefahren und -ziele – Kryptografische Algorithmen und Protokolle
Studien-/Prüfungsleistungen:	Klausur

Medienformen:	Overheadfolien, Beamerpräsentation, Software demonstration, Tafel, Handout
Literatur:	<p>SW-Design</p> <ul style="list-style-type: none"> • Uwe Schmidt: Softwaredesign, Vorlesungsunterlagen im Web: http://www.fh-wedel.de/~si/vorlesungen/softwaredesign/design.html • Gamma, Erich e. a.: Entwurfsmuster: Elemente wiederverwendbarer objektorientierter Software, Addison-Wesley, Bonn, 2001 (korrigierter Nachdruck) • Fowler, Martin; Scott, Kendall: UML Distilled. Applying The Standard Object Modelling Language, Addison-Wesley Longman, Amsterdam, 1997 • Bird, Richard: Introduction to Functional Programming using Haskell, 2nd Edition Prentice Hall, New Jersey, 1998 <p>Systemanalyse</p> <ul style="list-style-type: none"> • Häuslein, A.: Systemanalyse, vde-Verlag, 2004 • Krallmann, H., Frank, H., Gronau, N.: Systemanalyse im Unternehmen, Oldenbourg Verlag, 2002 • Rump, F. J.: Geschäftsprozessmodellierung auf der Basis ereignisgesteuerter Prozeßketten, Teubner, Stuttgart, 1999 • Oesterreich, B.: Objektorientierte Softwareentwicklung, München, 2004 <p>SW-Technik für Internetanwendungen</p> <ul style="list-style-type: none"> • Wöhr, Heiko: Webtechnologien, dpunkt.verlag, 2004 • Avci, O., Trittman, R., Mellis, W.: Web-Programmierung, Vieweg, 2003 • Langner, T.: Web-basierte Anwendungsentwicklung, Spektrum Akademischer Verlag, 2004 • Harold, E. R.: XML 1.1 Bible, Wiley Publishing, 2004 • Fuhrberg, K.: Internet-Sicherheit, Hanser, 2000

2.12 Informationssysteme

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Informationssysteme
Kürzel:	v35
Lehrveranstaltungen:	v351 Datenbanken 1 v350 Übung Datenbanken v352 Praktikum Datenbanken v353 Wissensbasierte Systeme
Semester:	3 (v351, v350), 5 (v352, v353)
Modulverantwortliche(r):	Sebastian Iwanowski
Dozent(in):	Hans-Detlef Gerhardt, Sebastian Iwanowski
Sprache:	deutsch
Zuordnung zum Curriculum:	Informatik (Bachelor): Pflicht, 3. Sem., 5. Sem.
Lehrform / SWS:	Vorlesung: 5 SWS, Gruppengröße: 50 - 115 Übung: 1 SWS, Gruppengröße: 20 Praktikum: 0 SWS, Gruppengröße: 1
Arbeitsaufwand:	Präsenzstudium: 56 Stunden, Eigenstudium: 184 Stunden
Kreditpunkte:	8
Voraussetzungen:	Einführung in die Informatik (Modul v15)
Lernziele / Kompetenzen:	<p>Beherrschen der Grundlagen der relationalen Datenbanktechnologie; Erwerb der Fähigkeit, selbständig einen Datenbankentwurfsprozess zu planen, eine relationale Datenbank unter Nutzung von SQL einzurichten und die Informationsverarbeitung mittels relationaler Datenbanksysteme unter Nutzung von SQL durchzuführen;</p> <p>Erwerb der Fähigkeit, mit einem Entwurfstool einen Datenbankentwurfsprozess durchzuführen und mittels SQL selbständig Anfragen an ein Datenbanksystem zu stellen;</p> <p>Erwerb praktischer Fertigkeiten durch die eigenständige Realisierung eines Datenbankentwurfsprozesses bis einschließlich der selbständigen Implementierung einer Datenbank für eine komplexe Datenverwaltungsaufgabe;</p> <p>Vermittlung eines grundsätzlichen Verständnisses der Technik der Wissensbasierten Systeme; Kenntnis sowohl der historischen Verbindung zur KI als auch der anwendungsorientierten Emanzipation von klassischen KI-Techniken, insbesondere für technische Anwendungen;</p> <p>Erlangen der Fähigkeit, die Eignung verschiedener wissensbasierter Techniken für vorgegebene Anwendungsbereiche zu erkennen.</p>

Inhalt:	<p>Datenbanken 1</p> <ul style="list-style-type: none"> • Einführung in die Datenbanktechnologie <ul style="list-style-type: none"> – Verwaltung von Informationen – Datenbanksysteme – Beispiel einer Datenbank • Datenbanksprache SQL - Einführung <ul style="list-style-type: none"> – Einführung in SQL – SQL-Datentypen, Konstanten, Variablen und Ausdrücke – Erstellen einer Datenbank • Datenbank-Abfrage mit SQL <ul style="list-style-type: none"> – Einführung in die vollständige SELECT - Anweisung – Komponenten der SELECT - Anweisung – Subquery, Join und Views • Datenbanksprache SQL- Einrichten der Datenbank <ul style="list-style-type: none"> – Anlegen von Tabellen und deren Erweiterungen – Indizes – Einfügen und Ändern von Daten in der Datenbank • Das Entity-Relationship-Datenmodell • Das Relationale Datenmodell <ul style="list-style-type: none"> – Relationenschemata und Datenabhängigkeiten – Funktionale Abhängigkeiten – Entwurfs-Theorie Relationaler Datenbank-Schemas • Relationale Datenbanksysteme <ul style="list-style-type: none"> – Das 3-Ebenen-Architekturkonzept – Transaktionskonzept – Relationale Operationen und Relationenalgebra – Komponenten eines DBMS's und Betrieb eines DBS's • Datenbank - Lebenszyklus <p>Wissensbasierte Systeme</p> <ul style="list-style-type: none"> • Grundlegende Prinzipien <ul style="list-style-type: none"> – Aufbau von Expertensystemen und wissensbasierten Systemen – Unterscheidung elementarer Begriffe wie Daten, Wissen, Constraints – Klassifizierung der Wissensqualität – Klassifizierung der Problemlösungsmechanismen • Historische Verbindung zur KI <ul style="list-style-type: none"> – Ziele des logischen Programmierens – Die KI als Motor heute allgemein anerkannter Paradigmen wie funktionaler, objektorientierter und agentenorientierter Programmierung • Suchstrategien <ul style="list-style-type: none"> – Breitensuche und Tiefensuche im Vergleich – Bestensuche – Anwendung auf Constraint Satisfaction Probleme • Modellbasierte Anwendungen in der Praxis <ul style="list-style-type: none"> – Diagnose- und Konfigurationsprobleme – Die General Diagnostic Engine (GDE) – Schnittstellen eines ATMS – Komponentenmodellierung
58	<ul style="list-style-type: none"> – Details zu einer GDE-Anwendung im Automobilbereich B_Inf1.0 (01.10.2004 bis 30.09.2007) • Weitere wissensbasierte Methoden <ul style="list-style-type: none"> – Symptombasiertes Schließen – Fallbasiertes Schließen – Neuronale Netze

Studien-/Prüfungsleistungen:	Klausur, Übungsabnahme
Medienformen:	Beamerpräsentation, Overheadfolien, Tafel, studentische Arbeit am Rechner
Literatur:	<p>Datenbanken 1</p> <ul style="list-style-type: none"> • Heuer, A. u. a.: Datenbanken kompakt, mitp-Verlag, Bonn 2003 • Meier, Andreas: Relationale Datenbanken Leitfaden für die Praxis, Springer Berlin 2004 • Vetter, M.: Aufbau betrieblicher Informationssysteme mittels konzeptioneller Datenmodellierung, Teubner, Stuttgart 1998 • Vossen, G.: Datenmodelle, Datenbanksprachen und Datenbank-Management-Systeme, Addison-Wesley, Bonn 2002, 4. Auflage <p>Wissensbasierte Systeme</p> <ul style="list-style-type: none"> • Günter Görz / Claus-Rainer Rollinger / Josef Schneeberger: Handbuch der Künstlichen Intelligenz, Oldenbourg 2000 (3. Auflage), ISBN 3-486-25049-3 • Stuart Russell / Peter Norvig: Künstliche Intelligenz: Ein moderner Ansatz, Pearson Studium 2004 (2. Auflage), ISBN 3-8273-7089-2 • Nguyen Huu Thuy / Peter Schnupp: Wissensverarbeitung und Expertensysteme, Oldenbourg 1989, ISBN 3-486-20699-0

2.13 ERP

Studiengang:	Bachelor Informatik
Modulbezeichnung:	ERP
Kürzel:	v36
Lehrveranstaltungen:	v360 ERP-Software, Übung ERP-Software v361 ERP-Systeme, Übung ERP-Systeme
Semester:	4 (v360), 5 (v361)
Modulverantwortliche(r):	Ulrich Raubach
Dozent(in):	Birger Wolter
Sprache:	deutsch
Zuordnung zum Curriculum:	Informatik (Bachelor), Pflicht, 4. Sem., 5. Sem.
Lehrform / SWS:	Vorlesung: 4 SWS, Gruppengröße: 40 - 75 Übung: 4 SWS, Gruppengröße: 20
Arbeitsaufwand:	Präsenzstudium: 74 Stunden, Eigenstudium: 166 Stunden
Kreditpunkte:	8
Voraussetzungen:	
Lernziele / Kompetenzen:	<p>Mit der Vorlesung ERP-Software sollen die Studierenden Wissen über die Realisierung der betriebswirtschaftlichen Grundfunktionalitäten innerhalb des SAP-Systems erwerben, das im Rahmen der Übung durch Fallbeispiele gefestigt wird.</p> <p>Der Programmiersprache ABAP kommt als Träger der betriebswirtschaftlichen Anwendungslogik in SAP-Systemen die nach wie vor grösste Bedeutung zu. Durch ABAP-Programmierkenntnisse werden die Studierenden befähigt, bestehende SAP-Systeme in ihrer Funktion zu verstehen und einfache Erweiterungen für spezielle Anwendungsaufgaben vorzunehmen.</p>
Inhalt:	<p>ERP-Software</p> <ul style="list-style-type: none"> • Finanzbuchhaltung (Sachkonten-, Debitoren-, Kreditorenbuchhaltung) • Kostenrechnung (Co) mit Gemeinkostencontrolling • Logistik <p>Übungen ERP-Software</p> <ul style="list-style-type: none"> • Fallstudien zu den Vorlesungsinhalten <p>ERP-Systeme, Übungen ERP-Systeme</p> <ul style="list-style-type: none"> • Einführung und erste Schritte • Werkzeuge der Entwicklungsumgebung • Grundlegende Konzepte • Datenbankzugriffe • Listenverarbeitung (Reports) • Gestaltung von Bildelementen (Dynpros) • Besonderheiten von Unternehmenssoftware • Ausblick: ABAP Objects und Business Server Pages
Studien-/Prüfungsleistungen:	Übungsabnahme, Klausur
Medienformen:	Rechner-/Beamer-Projektion der Vorlesungsmaterialien und der ERP-GUIs (Verbindung zum SAP Kompetenz-Center an der Universität Magdeburg) auf interaktiven Whiteboard

Literatur:	<ul style="list-style-type: none">● SAP-Dokumentationen im Internet: http://help.sap.com● Fallstudien des HCC der Uni Magdeburg● Vorlesungsfolien Birger Wolter (auf dem Handout-Server)● Maassen, Andre; Schoenen, Markus; Werr, Ina: Grundkurs SAP R/3, Vieweg● Kelch, Rainer: ABAP Objects, dpunkt-Verlag, 2002● Specht, Olaf; Wolter, Birger: Produktionslogistik mit PPS-Systemen, Kiehl, 2. Auflage, 1997
------------	--

2.14 Rechnernetze

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Rechnernetze
Kürzel:	v40
Lehrveranstaltungen:	v400 Rechnernetze v401 Praktikum Rechnernetze
Semester:	3 (v400), 4 (v401)
Modulverantwortliche(r):	Sebastian Iwanowski
Dozent(in):	Sebastian Iwanowski, Ilja Kaleck
Sprache:	deutsch
Zuordnung zum Curriculum:	Informatik (Bachelor): Pflicht, 3. Sem., 4. Sem.
Lehrform / SWS:	Vorlesung: 4 SWS, Gruppengröße: 25 Praktikum: 4 SWS, Gruppengröße: 10
Arbeitsaufwand:	Präsenzstudium: 74 Stunden, Eigenstudium: 166 Stunden
Kreditpunkte:	8
Voraussetzungen:	Informationstechnik 1 (in Modul v45) Unix + Internet & Unix-Übung (in Modul v30)
Lernziele / Kompetenzen:	<p>Verständnis für die Kommunikation zwischen Prozessen in Daten-netzen; Kenntnisse über den genauen Datenfluss in Internetprotokoll basierten Netzen, dazu Betrachtung der Arbeitsweise der aktuellen Internet-Protokoll Versionen (IPv4 und IPv6);</p> <p>Verständnis für den praktischen Aufbau und den Betrieb moder-ner Unternehmensnetze; Kenntnisse über die Eigenschaften aktueller Netztechnologien im Bereich der Lokalen Netze (LAN) und drahtlo-ser Netze (WLAN); Verständnis der Arbeitsweise der hier eingesetz-ten Koppelemente zum Aufbau größerer Netzstrukturen; Kenntniss-e über Arbeitsweise von Verzeichnisdiensten zur Verwaltung größerer Netze;</p> <p>Zusammenhang zwischen den praktisch eingesetzten Verfahren und theoretischen Verfahren aus der Mathematik</p> <p>Praktische Umsetzung der Theorie am eigenen System/Server, in-dividuelle Aneignung von weiterführenden Kenntnissen an aktueller Gerätetechnik, wie Router oder Firewall-Systemen.</p>

Inhalt:	<p>Rechnernetze</p> <ul style="list-style-type: none"> • Allgemeine Grundlagen und Begriffe <ul style="list-style-type: none"> – Netztopologien – Kurzeinführung SNA • Einfache Rechnernetze und Server-orientierte Netze <ul style="list-style-type: none"> – Datensicherungskonzepte und Speichernetze • ISO-OSI Referenzmodell <ul style="list-style-type: none"> – Kommunikation im Modell – Schichtenfunktionen im Detail • Internet-Architektur <ul style="list-style-type: none"> – Adressstrukturen und Umsetzungstechniken – UDP-/TCP-Kommunikation & Sockets – Beispiele für Anwendungsprotokolle – Grundlagen des IPv6, Netzstrukturen und Migrationstechniken • Lokale Netze <ul style="list-style-type: none"> – Kommunikation in IEEE-802 LANs – Ethernet-Technik (10/100/1Gbe/10GbE) • Koppелеlemente und Vermittlungstechniken <ul style="list-style-type: none"> – Repeater, Brücken, Router, Routingverfahren, – Switching-Technologie, Multi-Layer Switching – Virtuelle LANs und Quality-of-Services • Routing <ul style="list-style-type: none"> – Generelle Aufgabenstellung – Link-state-Verfahren – Distanzvektorverfahren – Hierarchisches Routing – Eingesetzte Protokolle in der Praxis – Broadcast-Verfahren – Multicast-Verfahren • Mobile Rechner <ul style="list-style-type: none"> – Allgemeine Prinzipien – Mobile IP – Mobilfunknetze <p>Praktikum Rechnernetze</p> <ul style="list-style-type: none"> • Einrichtung und Konfiguration grundlegender Kommunikationsprotokolle am eigenen System • Nutzung von Internetdienstprogramme mit genauer Betrachtung der dabei verwendeten Protokolle • Konfiguration und Arbeiten mit IPv6 • Einfache Benutzer- und Rechteverwaltung im Netz • Konfiguration grundlegender Internetserverdienste • Konfiguration und Einsatz von Techniken zur „Unix/Windows-Integration“ • Einrichten und Arbeiten mit aktuellen „Verzeichnisdiensten“ (NDS, ADS, LDAP) • Aufbau und Betrieb eines eigenen Wireless LAN • Protokollanalyse und Fehlersuche im LAN mit einem LAN-Analyser (LAN, WLAN)
---------	--

	<ul style="list-style-type: none"> • Aufbau einer Verkabelungsstrecke mit Vermessung unter Einsatz eines Kabelscanners • Bearbeitung ausgewählter Themenstellungen unter Einsatz von moderner Gerätetechnik (Cisco-Router, Pix-Firewall, Layer-3/-4 Switch, Streaming-Media Server).
Studien-/Prüfungsleistungen:	Klausur, Praktikumsabnahme
Medienformen:	Overheadfolien, Beamerpräsentation, Tafel, Handout, Arbeit am eigenen Praktikumsrechner, Netzgerätetechnik pro Student
Literatur:	<p>Rechnernetze</p> <ul style="list-style-type: none"> • Badach/Hoffmann: Technik der IP Netze, 1. Auflage, 2001, Hanser-Verlag • Tannenbaum, Andrew S.: Computer Netzwerke, 4. Auflage, Pearson-Studium • Stein, Erich: Taschenbuch Rechnernetze und Internet, 2. Auflage, Hanser-Verlag • Sikora, Axel: Technische Grundlagen der Rechnerkommunikation, Hanser/Fachbuchverlag-Leipzig • Franz-Joachim Kauffels: Lokale Netze (Band 1&2), 15. Auflage, MITP-Verlag • Kauffels, Franz-Joachim: Wireless LANs, MITP-Verlag <p>Praktikum Rechnernetze</p> <ul style="list-style-type: none"> • Rodriguez/Gatrell/Karas/Peschke: TCP/IP Tutorial and Technical Overview (PDF), IBM-Redbook Serie, Download unter http://www.redbooks.ibm.com/redbooks/ • Syngress: Cisco PIX Firewall MITP-Verlag • Andreas Tikart: Cisco Router - das Experimentierbuch, MITP-Verlag • IEEE Spezifikationen zu IEEE 802.11 • Internet RFCs, z. B. unter ftp://ftp.nordu.net/rfc

2.15 Einführung in die Informationsverarbeitung

2.15.1 bis 30.09.2008 (Pk)

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Einführung in die Informationsverarbeitung
Kürzel:	v45
Lehrveranstaltungen:	v450 Informationstechnik 1 u200 Digitaltechnik 1 v451 Praktikum Digitaltechnik
Semester:	1
Modulverantwortliche(r):	Wolfgang Ülzmann
Dozent(in):	Peter Poock-Haffmans, Wolfgang Ülzmann
Sprache:	deutsch
Zuordnung zum Curriculum:	Technische Informatik (Bachelor): Pflicht, 1. Sem. Informatik (Bachelor): Pflicht, 1. Sem.
Lehrform / SWS:	Vorlesung: 4 SWS, Gruppengröße: 35 - 150 Praktikum: 1 SWS, Gruppengröße: 20
Arbeitsaufwand:	Präsenzstudium: 47 Stunden, Eigenstudium: 73 Stunden
Kreditpunkte:	4
Voraussetzungen:	
Lernziele / Kompetenzen:	<p>Grundlegendes Verständnis für elementare Aspekte der Funktionalität von Rechnern, bezogen sowohl auf die informationstheoretischen Grundlagen als auch auf die praktische Implementierung benötigter Funktionselemente.</p> <p>Kenntnisse über wesentliche theoretische Grundlagen der Informationsverarbeitung; Verständnis für die praktische Umsetzung von Vorgängen der Informationsverarbeitung auf der Maschinenebene; Erfassen der rechnerinternen Abläufe am Beispiel einfacher Funktionsmodelle; Aufbauend auf einfachen Grundstrukturen schrittweises Verstehen immer komplexerer Verarbeitungsabläufe bis hin zu realen Strukturen.</p> <p>Beherrschen der formalen Beschreibungs-, Umformungs- und Darstellungsmethoden Boole'scher Funktionen; Kenntnisse über alle wesentlichen Grundelemente digitaler Systeme und deren Anwendung auf einfache technische Probleme; Erlangung der Kompetenz, einfache digitale Systeme zu entwerfen.</p> <p>Festigung des in der Vorlesung vermittelten Stoffes sowie die Übertragung des theoretischen Wissens in eine erlebbare, reale technische Umgebung.</p>

Inhalt:	<p>Informationstechnik 1</p> <ul style="list-style-type: none"> • Informationsdarstellung, Codierung <ul style="list-style-type: none"> – Begriffe, Konzepte der Codierung • Grundbegriffe der Schaltalgebra <ul style="list-style-type: none"> – Boole'sche Variable und Funktionen – Rechenwerke, Anwendungen • Zahlendarstellungen <ul style="list-style-type: none"> – Konvertierungen und Formate • Struktur und Funktion von Rechnern <ul style="list-style-type: none"> – CPU- und Speichermodelle – Maschinenbefehlsformate und Interpretation <p>Digitaltechnik 1</p> <ul style="list-style-type: none"> • Theoretische Grundlagen <ul style="list-style-type: none"> – Zweiwertige Funktionen – Boolesche Algebra – Disjunktive- und konjunktive Formen – Algebraische Minimierung – Karnaugh-Tafeln – Quine/McCluskey-Verfahren – Zeitverhalten realer Komponenten • Anwendungen <ul style="list-style-type: none"> – Schaltnetze: Encoder, Decoder – Schaltwerke: Flipflops, Zähler, Register <p>Praktikum Digitaltechnik</p> <ul style="list-style-type: none"> • Schaltnetzentwurf <ul style="list-style-type: none"> – Schaltnetz wird aus einer gegebenen Spezifikation formal entworfen. – Der Entwurf wird auf einem Steckbrett realisiert. – Die Schaltung wird auf Funktion und Einhaltung der Spezifikation überprüft. – Die Ergebnisse werden dokumentiert. • Schaltwerkentwurf <ul style="list-style-type: none"> – Schaltwerk (z. B. ein Zähler) wird aus einer gegebenen Spezifikation formal entworfen. – Der Entwurf wird auf einem Steckbrett realisiert. – Die Schaltung wird auf Funktion und Einhaltung der Spezifikation überprüft. – Die Ergebnisse werden dokumentiert.
Studien-/Prüfungsleistungen:	Vorlesung: Klausur Praktikum: Erfolgreiche Teilnahme
Medienformen:	Tafel, Beamerpräsentation, Overheadfolien, Handout

Literatur:	<p>Informationstechnik 1</p> <ul style="list-style-type: none">• Müller, Käser, et. al.: Technische Informatik 1, vdf-Hochschulverlag Zürich, 2003• Schiffmann, Schmitz: Technische Informatik 2, Grundlagen der Computertechnik, Springer-Verlag 1998• Heinrich, et. al.: Informations- und Kommunikationstechnik, Oldenbourg-Verlag, 1994• Hansen: Wirtschaftsinformatik 1, Verlag Lucius & Lucius, UTB 802, 8. Auflage, 2001 <p>Digitaltechnik 1</p> <ul style="list-style-type: none">• E. Prochaska: Digitaltechnik für Ingenieure, Oldenbourg• Lipp: Grundlagen der Digitaltechnik, Oldenbourg• Pernards: Digitaltechnik, Hüthig• Fricke: Digitaltechnik, Vieweg• Borgmeyer: Grundlagen der Digitaltechnik, Hanser• Siemers, Sikora: Taschenbuch Digitaltechnik, Hanser• Marcovitz: Introduction to Logic Design, McGrawhill
------------	---

2.15.2 ab 01.10.2008 (Saw)

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Einführung in die Informationsverarbeitung
Kürzel:	v45
Lehrveranstaltungen:	v450 Informationstechnik 1 u200 Digitaltechnik 1 v451 Praktikum Digitaltechnik
Semester:	1
Modulverantwortliche(r):	Wolfgang Ülzmann
Dozent(in):	Sergei Sawitzki, Wolfgang Ülzmann
Sprache:	deutsch
Zuordnung zum Curriculum:	Technische Informatik (Bachelor): Pflicht, 1. Sem. Informatik (Bachelor): Pflicht, 1. Sem.
Lehrform / SWS:	Vorlesung: 4 SWS, Gruppengröße: 35 - 150 Praktikum: 1 SWS, Gruppengröße: 20
Arbeitsaufwand:	Präsenzstudium: 47 Stunden, Eigenstudium: 73 Stunden
Kreditpunkte:	4
Voraussetzungen:	
Lernziele / Kompetenzen:	<p>Grundlegendes Verständnis für elementare Aspekte der Funktionalität von Rechnern, bezogen sowohl auf die informationstheoretischen Grundlagen als auch auf die praktische Implementierung benötigter Funktionselemente.</p> <p>Kenntnisse über wesentliche theoretische Grundlagen der Informationsverarbeitung; Verständnis für die praktische Umsetzung von Vorgängen der Informationsverarbeitung auf der Maschinenebene; Erfassen der rechnerinternen Abläufe am Beispiel einfacher Funktionsmodelle; Aufbauend auf einfachen Grundstrukturen schrittweises Verstehen immer komplexerer Verarbeitungsabläufe bis hin zu realen Strukturen.</p> <p>Beherrschen der formalen Beschreibungs-, Umformungs- und Darstellungsmethoden Boole'scher Funktionen; Kenntnisse über alle wesentlichen Grundelemente digitaler Systeme und deren Anwendung auf einfache technische Probleme; Erlangung der Kompetenz, einfache digitale Systeme zu entwerfen.</p> <p>Festigung des in der Vorlesung vermittelten Stoffes sowie die Übertragung des theoretischen Wissens in eine erlebbare, reale technische Umgebung.</p>

Inhalt:	<p>Informationstechnik 1</p> <ul style="list-style-type: none"> • Informationsdarstellung, Codierung <ul style="list-style-type: none"> – Begriffe, Konzepte der Codierung • Grundbegriffe der Schaltalgebra <ul style="list-style-type: none"> – Boole'sche Variable und Funktionen – Rechenwerke, Anwendungen • Zahlendarstellungen <ul style="list-style-type: none"> – Konvertierungen und Formate • Struktur und Funktion von Rechnern <ul style="list-style-type: none"> – CPU- und Speichermodelle – Maschinenbefehlsformate und Interpretation <p>Digitaltechnik 1</p> <ul style="list-style-type: none"> • Einleitung • Digitale Systeme • Schaltalgebra <ul style="list-style-type: none"> – Entstehungsgeschichte – Boolesche Algebra – Schaltfunktionen – Operatorensysteme – Normalformen und Dualitätsprinzip • Schaltnetze <ul style="list-style-type: none"> – Darstellung – Vereinfachung (KV-Diagramme, QMCV, BDDs) – Analyse (Funktion, Komplexität, Zeitverhalten) – Synthese und Realisierung – Beispiele • Speicherelemente <p>Praktikum Digitaltechnik</p> <ul style="list-style-type: none"> • Schaltnetzentwurf <ul style="list-style-type: none"> – Schaltnetz wird aus einer gegebenen Spezifikation formal entworfen. – Der Entwurf wird auf einem Steckbrett realisiert. – Die Schaltung wird auf Funktion und Einhaltung der Spezifikation überprüft. – Die Ergebnisse werden dokumentiert. • Schaltwerkentwurf <ul style="list-style-type: none"> – Schaltwerk (z. B. ein Zähler) wird aus einer gegebenen Spezifikation formal entworfen. – Der Entwurf wird auf einem Steckbrett realisiert. – Die Schaltung wird auf Funktion und Einhaltung der Spezifikation überprüft. – Die Ergebnisse werden dokumentiert.
Studien-/Prüfungsleistungen:	Vorlesung: Klausur Praktikum: Erfolgreiche Teilnahme
Medienformen:	Tafel, Beamerpräsentation, Overheadfolien, Handout

Literatur:	<p>Informationstechnik 1</p> <ul style="list-style-type: none">• Müller, Käser, et. al.: Technische Informatik 1, vdf-Hochschulverlag Zürich, 2003• Schiffmann, Schmitz: Technische Informatik 2, Grundlagen der Computertechnik, Springer-Verlag 1998• Heinrich, et. al.: Informations- und Kommunikationstechnik, Oldenbourg-Verlag, 1994• Hansen: Wirtschaftsinformatik 1, Verlag Lucius & Lucius, UTB 802, 8. Auflage, 2001 <p>Digitaltechnik 1</p> <ul style="list-style-type: none">• HOFFMANN, Dirk: Grundlagen der technischen Informatik, Carl Hanser Verlag 2007• SCHIFFMANN, Wolram; SCHMITZ, Robert: Technische Informatik, in 3 Bänden. 3. Auflage Springer Verlag, 1996• BEUTH, Klaus: Elektronik 4. Digitaltechnik. 13. Auflage Vogel Verlag und Druck 2003
------------	---

2.16 Technik der Informationsverarbeitung

2.16.1 bis 30.09.2008 (Pk)

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Technik der Informationsverarbeitung
Kürzel:	v46
Lehrveranstaltungen:	v460 Informationstechnik 2 v461 Rechnerstrukturen v462 Digitaltechnik 2
Semester:	2 (v461, v462), 3 (v460)
Modulverantwortliche(r):	Wolfgang Ülzmann
Dozent(in):	Peter Pooock-Haffmans, Wolfgang Ülzmann
Sprache:	deutsch
Zuordnung zum Curriculum:	Informatik (Bachelor): Pflicht, 2. Sem., 3. Sem.
Lehrform / SWS:	Vorlesung: 6 SWS, Gruppengröße: 25 - 65
Arbeitsaufwand:	Präsenzstudium: 56 Stunden, Eigenstudium: 124 Stunden
Kreditpunkte:	6
Voraussetzungen:	Einführung in die Informationsverarbeitung (Modul v45)
Lernziele / Kompetenzen:	<p>Verständnis für die Abläufe beim Betrieb realer Rechnerkomponenten, und für die Probleme beim abstrakten Entwurf digitaler Systeme mit modernen CAE-Methoden.</p> <p>Verständnis der rechnerinternen Interpretationsvorgänge der Maschinenbefehle; Kompetenz zur effizienten Anwendung von beliebigen Befehlsstrukturen höherer Sprachebenen bezogen auf ihre Umsetzung im Rechner; Erkennung der Bedeutung von Datenübertragungsverfahren und Protokollen; Vernetzung und Vervollständigung des Wissens über interne Funktionsabläufe in der Prozessorperipherie.</p> <p>Durchdringung grundlegender Rechnerarchitekturkonzepte, die beschreiben, wie verschiedene Baugruppen von Rechnern zusammenarbeiten und wie sich unterschiedliche Rechnersysteme voneinander unterscheiden. Erkennen der Bedeutung des Zusammenwirkens aller beteiligten Hardware- und Softwarekonzepte; Erkennung des Zusammenspiels aller Funktionselemente von Rechnern mit ihren typischen Systemeigenschaften und deren Abbildung auf ein bestimmtes Architekturmodell; Verständnis für Ansätze zur Steigerung der Systemleistung; Erwerb des theoretischen Wissens, das für den abstrakten Entwurf digitaler Systeme benötigt wird.</p> <p>Durchdringung des vollständigen Designflow einer modernen Schaltungsentwicklung; Kenntnisse über Hardwarebeschreibungssprachen am Beispiel von VHDL.</p> <p>Tiefgreifendes Verständnis für alle Konzepte der Schaltungsentwicklung, beginnend mit der funktionalen Beschreibung bis zum Download in den Schaltkreis.</p>

Inhalt:	<p>Informationstechnik 2</p> <ul style="list-style-type: none"> • Vertiefung Rechenwerke <ul style="list-style-type: none"> – Schieberegister, Parallele und serielle Strukturen • Kommunikationskanäle <ul style="list-style-type: none"> – Verbindungskonzepte – Bussysteme • Adressierungstechniken <ul style="list-style-type: none"> – absolut, relativ, indiziert, indirekt – Speicherverwaltung mit MMU-Strukturen • Erweiterte Befehlsbearbeitung <ul style="list-style-type: none"> – Maschineninterpretation – Vollständige Mikroprogramme • Ereignisbearbeitung <ul style="list-style-type: none"> – Polling, Daisy-Chaining • Ein-/Ausgabetechnik <ul style="list-style-type: none"> – Programmgesteuerte EA – Direct Memory Access • Technologie Externer Speicher <ul style="list-style-type: none"> – Magnetomotorische Konzepte • Elektronische Speichermedien <p>Rechnerstrukturen</p> <ul style="list-style-type: none"> • Aspekte der Rechnerarchitekturen und Begriffe • Hierarchisches Schichtenmodell • Hardwarekomponenten und Operationsprinzipien • Architekturkonzepte nach von Neumann <ul style="list-style-type: none"> – Ablaufkontrolle, Speicherorganisation – Bedeutung der Cache-Speicher • Mikroprogrammierung <ul style="list-style-type: none"> – Vertikale Verlagerung – Virtuelle Maschinen – Nanoprogrammierung • RISC-Konzepte <ul style="list-style-type: none"> – Befehlssätze – Probleme beim Pipelining – Leistungsbewertung • Parallelität und Nicht Sequentielle Architekturen <ul style="list-style-type: none"> – Parallelismus und Nutzbarkeit • Datenflussorientierte Systeme <p>Digitaltechnik 2</p> <ul style="list-style-type: none"> • Programmierbare Bausteine <ul style="list-style-type: none"> – PAL-Strukturen – CPLDs – FPGAs • CAE-Methoden der Digitaltechnik <ul style="list-style-type: none"> – Allgemeiner Designflow – Hardwarebeschreibungssprachen – Einführung in VHDL
---------	---

Studien-/Prüfungsleistungen:	Klausur
Medienformen:	Tafel, Beamerpräsentation, Overheadfolien, Handout
Literatur:	Informationstechnik 2 <ul style="list-style-type: none">• Müller, Käser, et. al.: Technische Informatik 1, vdf-Hochschulverlag Zürich, 2003• Schiffmann, Schmitz: Technische Informatik 2, Grundlagen der Computertechnik, Springer-Verlag, 1998• Heinrich, et. al.: Informations- und Kommunikationstechnik, Oldenbourg-Verlag, 1994

- Hansen:
Wirtschaftsinformatik 1,
Verlag Lucius & Lucius, UTB 802, 8. Auflage, 2001

Rechnerstrukturen

- Märtin:
Einführung in die Rechnerarchitektur,
Fachbuchverlag Leipzig, 2003
- Oberschelp, Gossen:
Rechneraufbau und Rechnerstrukturen,
Verlag Oldenbourg, 1998
- van de Goor:
Computer Architecture & Design,
Verlag Addison Wesley, 1989
- Müller-Schloer, Schmitter:
RISC-Workstation Architekturen,
Verlag Springer, 1991

Digitaltechnik 2

- Herrmann, Müller:
ASIC-Entwurf und Test,
Fachbuchverlag Leipzig
- Jorke:
Rechnergestützter Entwurf digitaler Schaltungen,
Fachbuchverlag Leipzig/Hanser
- Hertwig, Brück:
Entwurf digitaler Systeme,
Hanser
- Wuttke, Henke:
Schaltssysteme,
Pearson Studium
- Siemers:
Hardwaremodellierung,
Hanser
- Jansen:
Handbuch der Electronic Design Automation,
Hanser
- Reichardt, Schwarz:
VHDL-Synthese,
Oldenbourg
- Molitor, Ritter:
VHDL-Eine Einführung,
Pearson Studium
- Heinkel, ...:
The VHDL Reference,
Wiley
- Brown, Vranesic:
Fundamentals of Digital Logic with VHDL Design,
McGrawHill
- Rushton:
VHDL for Logic Synthesis,
Wiley
- Sjöholm, Lindh:
VHDL for Designers,
Pearson
- Perry:
VHDL Programming by Example

2.16.2 ab 01.10.2008 (Saw)

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Technik der Informationsverarbeitung
Kürzel:	v46
Lehrveranstaltungen:	v460 Informationstechnik 2 v461 Rechnerstrukturen v462 Digitaltechnik 2
Semester:	2 (v461, v462), 3 (v460)
Modulverantwortliche(r):	Wolfgang Ülzmann
Dozent(in):	Sergei Sawitzki, Wolfgang Ülzmann
Sprache:	deutsch
Zuordnung zum Curriculum:	Informatik (Bachelor): Pflicht, 2. Sem., 3. Sem.
Lehrform / SWS:	Vorlesung: 6 SWS, Gruppengröße: 25 - 65
Arbeitsaufwand:	Präsenzstudium: 56 Stunden, Eigenstudium: 124 Stunden
Kreditpunkte:	6
Voraussetzungen:	Einführung in die Informationsverarbeitung (Modul v45)
Lernziele / Kompetenzen:	<p>Verständnis für die Abläufe beim Betrieb realer Rechnerkomponenten, und für die Probleme beim abstrakten Entwurf digitaler Systeme mit modernen CAE-Methoden.</p> <p>Verständnis der rechnerinternen Interpretationsvorgänge der Maschinenbefehle; Kompetenz zur effizienten Anwendung von beliebigen Befehlsstrukturen höherer Sprachebenen bezogen auf ihre Umsetzung im Rechner; Erkennung der Bedeutung von Datenübertragungsverfahren und Protokollen; Vernetzung und Vervollständigung des Wissens über interne Funktionsabläufe in der Prozessorperipherie.</p> <p>Durchdringung grundlegender Rechnerarchitekturkonzepte, die beschreiben, wie verschiedene Baugruppen von Rechnern zusammenarbeiten und wie sich unterschiedliche Rechnersysteme voneinander unterscheiden. Erkennen der Bedeutung des Zusammenwirkens aller beteiligten Hardware- und Softwarekonzepte; Erkennung des Zusammenspiels aller Funktionselemente von Rechnern mit ihren typischen Systemeigenschaften und deren Abbildung auf ein bestimmtes Architekturmodell; Verständnis für Ansätze zur Steigerung der Systemleistung; Erwerb des theoretischen Wissens, das für den abstrakten Entwurf digitaler Systeme benötigt wird.</p> <p>Durchdringung des vollständigen Designflow einer modernen Schaltungsentwicklung; Kenntnisse über Hardwarebeschreibungssprachen am Beispiel von VHDL.</p> <p>Tiefgreifendes Verständnis für alle Konzepte der Schaltungsentwicklung, beginnend mit der funktionalen Beschreibung bis zum Download in den Schaltkreis.</p>

Inhalt:	<p>Informationstechnik 2</p> <ul style="list-style-type: none">• Vertiefung Rechenwerke<ul style="list-style-type: none">– Schieberegister, Parallele und serielle Strukturen• Kommunikationskanäle<ul style="list-style-type: none">– Verbindungskonzepte– Bussysteme• Adressierungstechniken<ul style="list-style-type: none">– absolut, relativ, indiziert, indirekt– Speicherverwaltung mit MMU-Strukturen• Erweiterte Befehlsbearbeitung<ul style="list-style-type: none">– Maschineninterpretation– Vollständige Mikroprogramme• Ereignisbearbeitung<ul style="list-style-type: none">– Polling, Daisy-Chaining• Ein-/Ausgabetechnik<ul style="list-style-type: none">– Programmgesteuerte EA– Direct Memory Access• Technologie Externer Speicher<ul style="list-style-type: none">– Magnetomotorische Konzepte• Elektronische Speichermedien <p>Rechnerstrukturen</p> <ul style="list-style-type: none">• Aspekte der Rechnerarchitekturen und Begriffe• Hierarchisches Schichtenmodell• Hardwarekomponenten und Operationsprinzipien• Architekturkonzepte nach von Neumann<ul style="list-style-type: none">– Ablaufkontrolle, Speicherorganisation– Bedeutung der Cache-Speicher• Mikroprogrammierung<ul style="list-style-type: none">– Vertikale Verlagerung– Virtuelle Maschinen– Nanoprogrammierung• RISC-Konzepte<ul style="list-style-type: none">– Befehlssätze– Probleme beim Pipelining– Leistungsbewertung• Parallelität und Nicht Sequentielle Architekturen<ul style="list-style-type: none">– Parallelismus und Nutzbarkeit• Datenflussorientierte Systeme
---------	--

	<p>Digitaltechnik 2</p> <ul style="list-style-type: none"> • Einleitung <ul style="list-style-type: none"> – Lernziele – Organisatorisches – Literatur – Bezeichnungen und Konventionen – Einordnung und historische Entwicklung – Voraussetzungen • Schaltwerke <ul style="list-style-type: none"> – Einleitung und Grundbegriffe, Definitionen – Speicherelemente – Analyse – Synthese – Realisierung, Beispiele • VHDL <ul style="list-style-type: none"> – Einleitung – Modellaufbau – Basiselemente – Sprachkonstrukte – Objekte – Entwurfseinheiten – Modellierungstechniken – Simulation – Weiterführende Konzepte – Synthese
Studien-/Prüfungsleistungen:	Klausur
Medienformen:	Tafel, Beamerpräsentation, Overheadfolien, Handout
Literatur:	<p>Informationstechnik 2</p> <ul style="list-style-type: none"> • Müller, Käser, et. al.: Technische Informatik 1, vdf-Hochschulverlag Zürich, 2003 • Schiffmann, Schmitz: Technische Informatik 2, Grundlagen der Computertechnik, Springer-Verlag, 1998 • Heinrich, et. al.: Informations- und Kommunikationstechnik, Oldenbourg-Verlag, 1994

- Hansen:
Wirtschaftsinformatik 1,
Verlag Lucius & Lucius, UTB 802, 8. Auflage, 2001

Rechnerstrukturen

- Märtin:
Einführung in die Rechnerarchitektur,
Fachbuchverlag Leipzig, 2003
- Oberschelp, Gossen:
Rechneraufbau und Rechnerstrukturen,
Verlag Oldenbourg, 1998
- van de Goor:
Computer Architecture & Design,
Verlag Addison Wesley, 1989
- Müller-Schloer, Schmitter:
RISC-Workstation Architekturen,
Verlag Springer, 1991

Digitaltechnik 2

- REICHARDT, Jürgen; SCHWARZ, Bernd:
VHDL-Synthese: Entwurf digitaler Schaltungen und Systeme,
4. Auflage
Oldenbourg Verlag 2007
- HOFFMANN, Dirk:
Grundlagen der technischen Informatik,
Carl Hanser Verlag 2007
- SCHIFFMANN, Wolram; SCHMITZ, Robert:
Technische Informatik, in 3 Bänden. 3. Auflage
Springer Verlag, 1996
- ASHENDEN, Peter:
The Designer's Guide to VHDL. 3rd edition
Elsevier 2008
- BEUTH, Klaus:
Elektronik 4. Digitaltechnik. 13. Auflage
Vogel Verlag und Druck 2003

2.17 Computergrafik

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Computergrafik
Kürzel:	v50
Lehrveranstaltungen:	v500 Praktikum Computergrafik 1 v501 Computergrafik 1 v502 Praktikum Bildbearbeitung v503 Bildbearbeitung
Semester:	4(v500, v501), 5 (v502, v503)
Modulverantwortliche(r):	Christian-Arved Bohn
Dozent(in):	Christian-Arved Bohn
Sprache:	deutsch
Zuordnung zum Curriculum:	Informatik (Bachelor): Pflicht, 4. Sem., 5. Sem. Medieninformatik (Bachelor): Pflicht, 4. Sem., 5. Sem.
Lehrform / SWS:	Vorlesung: 3 SWS, Gruppengröße: 140 Praktikum: 3 SWS, Gruppengröße: 20
Arbeitsaufwand:	Präsenzstudium: 56 Stunden, Eigenstudium: 124 Stunden
Kreditpunkte:	6
Voraussetzungen:	
Lernziele / Kompetenzen:	<p>Lernziele sind die grundlegenden Techniken der Computergrafik, gegliedert in zwei große Bereiche, nämlich dem der dreidimensionalen Bildgenerierung („Computergrafik 1“) und dem der zweidimensionalen Bildbearbeitung. Eine Brücke zwischen beiden Bereichen wird durch das relativ neue Feld des Non-Photorealistic Rendering geschaffen.</p> <p>In den Veranstaltungen zum Thema „Computergrafik 1“ wird besonderer Wert auf die enge Verknüpfung zwischen der Vorlesung und dem Praktikum gelegt. Hierdurch können theoretische und praktische Aspekte der Vorlesung auf einfachere Weise erlernt werden, das Trainieren der sozialen Kompetenz der Studierenden wird durch Gruppenarbeit im Praktikum gefördert.</p> <p>Die Bildbearbeitung fügt diesem Modul Grundwissen über ein Gebiet hinzu, das derzeit in der Medieninformatik immer mehr an Bedeutung gewinnt.</p> <p>Auch hier wird auf den Bezug zur praktischen Realisierung Wert gelegt, da diese Thematik sehr gut durch praktische Erfahrungen erlernt werden kann.</p>

Inhalt:	<p>Computergrafik 1</p> <ul style="list-style-type: none"> • Zusammenhang „Bild“ und „Pixel“ • pixelbasierte 2D-Algorithmen (Linien, Flächen, Clipping) • Grundlagen geometrischer Modellierung • Vektoren und lineare Algebra • Bilderzeugung <ul style="list-style-type: none"> – Strahlverfolgung für Projektion und Lichtsimulation – Raycasting, Raytracing – lokale Beleuchtungsmodelle (Phong, Blinn) – Ansatz der „Rendering Pipeline“ – „Szenegraph“ – Projektion – 2D-Füllverfahren – OpenGL • Spezielle Techniken des Rendering <ul style="list-style-type: none"> – Texturierung – Echtzeit-Schattengenerierung – Algorithmen für aktuelle Computergrafik-Hardware <p>Bildbearbeitung</p> <ul style="list-style-type: none"> • Begriffe „Bild, Pixel, Farbdarstellung“ • Affine Transformationen im Bildraum • Histogrammbasierte Operationen • Zusammenhang Frequenz und Bildbereich • Fouriertransformation, Filtertechniken • Kombination/Anwendung der erlernten Verfahren für konkrete Problemstellungen • Spezialalgorithmen der Bildbearbeitung • Techniken des Non-Photorealistic Rendering • 2D-Bearbeitung dreidimensionaler Bilder
Studien-/Prüfungsleistungen:	Mündliche Prüfung, Übungsabnahme
Medienformen:	Tafel, Overheadfolien, Beamerpräsentation, Handout, studentische Arbeit am Rechner

Literatur:	<p>Computergrafik 1</p> <ul style="list-style-type: none">• D. Hearn, M. P. Baker: Computer Graphics with OpenGL, Pearson Education International, 2004• J. Encarnacao, W. Straßer, R. Klein: Graphische Datenverarbeitung 1, Oldenbourg Verlag GmbH, 1996• M. E. Mortenson: Mathematics for Computer Graphics Applications, Industrial Press Inc., 1999• T. Möller, E. Haines: Real-Time Rendering, A K Peters, Ltd., 1999 <p>Bildbearbeitung</p> <ul style="list-style-type: none">• J. C. Russ: The Image Processing Handbook, CRC Press LLC, 1999• C. Bunks: Grokking the GIMP, new Riders Publishing, 2000• B. Jähne: Digitale Bildverarbeitung, Springer, 2002• acm Transactions on Graphics (Proceedings of siggraph), ACM, 1998-2004
------------	--

2.18 Betriebswirtschaftslehre

2.18.1 bis 31.03.2009 (Bau)

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Betriebswirtschaftslehre
Kürzel:	v55
Lehrveranstaltungen:	v550 Allgemeine Betriebswirtschaftslehre v551 Rechnungswesen
Semester:	1 (v550), 2 (v551)
Modulverantwortliche(r):	Sabine Baumann
Dozent(in):	Sabine Baumann, Birger Wolter
Sprache:	deutsch
Zuordnung zum Curriculum:	Informatik (Bachelor): Pflicht, 1. und 2. Sem. Technische Informatik (Bachelor): Pflicht, 3. und 4. Sem.
Lehrform / SWS:	Vorlesung mit zahlreichen Praxisbeispielen und eingebetteten Übungsaufgaben: 6 SWS, Gruppengröße 25-200
Arbeitsaufwand:	Präsenzstudium 56 Stunden, Eigenstudium 124 Stunden
Kreditpunkte:	6
Voraussetzungen:	
Lernziele / Kompetenzen:	<p>Die wesentlichen Teilgebiete der BWL kennen und erläutern können. Die Begriffe der betriebswirtschaftlichen Teilgebiete kennen, abgrenzen und erläutern können.</p> <p>Zentrale Methoden der BWL kennen und erläutern können; Übertragung und Anwendung der Methoden auf einfach-strukturierte, praktische Problemstellungen.</p> <p>Kenntnis der grundlegenden Konzepte der Finanzbuchhaltung und Kostenrechnung sowie ihrer Umsetzung in kaufmännischer Standardsoftware;</p> <p>Fähigkeit zum Einsatz dieser Konzepte zur Durchführung und Gestaltung entsprechender Geschäftsprozesse.</p>

Inhalt:	<p>Allgemeine Betriebswirtschaftslehre</p> <ul style="list-style-type: none">• Grundlagen und Rahmenbedingungen<ul style="list-style-type: none">– Gegenstand und Betrachtungsebenen der Betriebswirtschaftslehre– Gesellschaftliches, wirtschaftliches und rechtliches Umfeld des Unternehmens– Ziele des Unternehmens• Beschaffung und Materialwirtschaft<ul style="list-style-type: none">– Grundlagen– Gestaltung des Beschaffungsvorgangs– Lagerhaltung und Transport– Umweltorientierte Materialwirtschaft und Entsorgung• Produktionswirtschaft<ul style="list-style-type: none">– Grundlagen– Gestaltung der Rahmenbedingungen– Produktionsgestaltung– Produktions- und Kostentheorie• Marketing und Vertrieb<ul style="list-style-type: none">– Grundlagen– Marktforschung– Zielfestlegung und Strategie– Produktpolitik– Konditionenpolitik– Distributionspolitik– Kommunikationspolitik– Marketing-Mix
---------	---

	<ul style="list-style-type: none"> • Personalwirtschaft <ul style="list-style-type: none"> – Begriff und Aufgabenstellung – Gestaltungsfelder – Führung • Organisation <ul style="list-style-type: none"> – Begriff und Ziele – Gestaltungsfelder – Organisationskontext und organisatorischer Wandel – Organisationstheoretische Ansätze • Rechnungswesen <ul style="list-style-type: none"> – Teilgebiete, Aufgaben und Begriffe – Jahresabschluss – Kosten- und Leistungsrechnung • Investition und Finanzierung <ul style="list-style-type: none"> – Investitionsprozess – Investitionsrechnung – Grundlagen der Finanzierung <p>Rechnungswesen</p> <ul style="list-style-type: none"> • Praxis der Geschäftsbuchführung nach dem IKR <ul style="list-style-type: none"> – Buchung von Einkaufs- und Verkaufsvorgängen – Buchungen im Zahlungs- und Finanzbereich – Buchungen im Sachanlagenbereich • Jahresabschluss der Unternehmung • Industrielle Kosten- und Leistungsrechnung <ul style="list-style-type: none"> – Kostenarten-, Kostenstellen- und Kostenträgerrechnung – Deckungsbeitragsrechnung als Teilkostenrechnung • Einführung in die flexible Plankostenrechnung
Studien-/Prüfungsleistungen:	Klausur
Medienformen:	Beamerpräsentation, Tafel, Overheadfolien, Handout

Literatur:	<p>Allgemeine Betriebswirtschaftslehre</p> <ul style="list-style-type: none">• Weber, W. (2003): Einführung in die Betriebswirtschaftslehre, 5. Auflage (Heidelberg: Gabler)• Thommen, J.-P./Achleitner A.-K. (2003): Allgemeine Betriebswirtschaftslehre, Umfassende Einführung aus managementorientierter Sicht, 4. Auflage (Heidelberg: Gabler)• Thommen, J.-P./Achleitner A.-K./Poech, A. (2004): Allgemeine Betriebswirtschaftslehre - Arbeitsbuch, 4. Auflage (Heidelberg: Gabler) <p>Rechnungswesen</p> <ul style="list-style-type: none">• Schmolke, Siegfried; Deitermann, Manfred: Industrielles Rechnungswesen - IKR, Winklers Verlag, 2004 (32. Auflage)• Bussiek, Jürgen; Ehrmann, Harald: Kompendium der praktischen Betriebswirtschaft, Buchführung, Kiehl Verlag, 2004 (8. Auflage)• Olfert, Klaus: Kompendium der praktischen Betriebswirtschaft, Kostenrechnung, Kiehl Verlag, 2003 (13. Auflage)
------------	---

2.18.2 ab 01.04.2009 (Gh)

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Betriebswirtschaftslehre
Kürzel:	v55
Lehrveranstaltungen:	v550 Allgemeine Betriebswirtschaftslehre v551 Rechnungswesen
Semester:	1 (v550), 2 (v551)
Modulverantwortliche(r):	Gunnar Harms
Dozent(in):	Gunnar Harms, Birger Wolter
Sprache:	deutsch
Zuordnung zum Curriculum:	Informatik (Bachelor): Pflicht, 1. und 2. Sem. Technische Informatik (Bachelor): Pflicht, 3. und 4. Sem.
Lehrform / SWS:	Vorlesung mit zahlreichen Praxisbeispielen und eingebetteten Übungsaufgaben: 6 SWS, Gruppengröße 25-200
Arbeitsaufwand:	Präsenzstudium 56 Stunden, Eigenstudium 124 Stunden
Kreditpunkte:	6
Voraussetzungen:	
Lernziele / Kompetenzen:	<p>Nach Abschluss der Veranstaltung verfügen die Studierenden über ein fundiertes Wissen in ausgewählten Teilbereichen der Betriebswirtschaftslehre. Die Studierenden erlernen Grundtatbestände der Allgemeinen Betriebswirtschaftslehre, beginnend vom Erkenntnisobjekt dieser wissenschaftlichen Disziplin, über die zu fällenden konstitutiven Entscheidungen, bis hin zu den diversen betriebswirtschaftlichen Funktionen (z. B. Beschaffung, Produktion und Absatz) innerhalb eines Betriebes. Letztere stehen im Mittelpunkt der Veranstaltung.</p> <p>Durch zahlreiche Übungen wird das Verständnis für die betriebswirtschaftlichen Prozesse und deren Zusammenhänge gefestigt sowie das eigenständige Arbeiten gefördert.</p> <p>Kenntnis der grundlegenden Konzepte der Finanzbuchhaltung und Kostenrechnung sowie ihrer Umsetzung in kaufmännischer Standardsoftware;</p> <p>Fähigkeit zum Einsatz dieser Konzepte zur Durchführung und Gestaltung entsprechender Geschäftsprozesse.</p>

Medienformen:	Beamerpräsentation, Tafel, Overheadfolien, Handout
---------------	--

Literatur:	<p>Allgemeine Betriebswirtschaftslehre</p> <ul style="list-style-type: none"> • Becker, H. P.: Investition und Finanzierung, 1. Aufl., Wiesbaden 2007 • Bernecker, M.: Grundlagen der Betriebswirtschaftslehre, 2. Aufl., Köln 2008 • Blom, H.; Beer, T.; Seidenberg, U.; Silber, H.: Produktionswirtschaft, 4. Aufl., Herne 2008 • Camphausen, B.: Strategisches Management, 2. Aufl., München 2007. • Däumler, K.-D.: Grundlagen der Investitions- und Wirtschaftlichkeitsrechnung, 11. Aufl., Berlin 2003 • Gienke, H.; Kämpf, R.: Handbuch Produktion: Innovatives Produktionsmanagement: Organisation, Konzepte, Controlling, München 2007 • Hansmann, K.-W.: Industrielles Management, 7. Aufl., München/Wien 2001 • Jung, H.: Allgemeine Betriebswirtschaftslehre, 11. Aufl., München 2009 • Melzer-Ridinger, R.: Materialwirtschaft und Einkauf, 5. Aufl., München 2008 • Olfert, K.; Reichel, C.: Investition, 11. Aufl., Ludwigshafen (Rhein)/Kiel 2009 • Schneeweiß, C.: Einführung in die Produktionswirtschaft, 8. Aufl., Berlin/Heidelberg/ New York, 2002 • Schneider, D.: Investition, Finanzierung und Besteuerung, 7. Aufl., Wiesbaden 1992 • Schneider, D.: Unternehmensführung und strategisches Controlling, 2. Aufl., München 2000 • Schulte, C.: Logistik, 3. Aufl., München 1999 • Specht, O.; Schmitt, U.: Betriebswirtschaftslehre für Ingenieure + Informatiker, 5. überarb. Aufl., München/Wien 2000 • Thommen, J.-P.; Achleitner, A.-K.: Allgemeine Betriebswirtschaftslehre, 5. Aufl., Wiesbaden 2006 • Vahs, D.; Schäfer-Kunz, J.: Einführung in die Betriebswirtschaftslehre, Stuttgart 2007 • Vollmer, T: Einführung in die Betriebswirtschaftslehre, in: Grundlagen der Betriebswirtschaft, Camphausen, B. (Hrsg.), München 2008 • Weber, W.; Kabst, R.: Einführung in die Betriebswirtschaftslehre, 7. überarb. Aufl., Wiesbaden 2009 • Wiendahl, H.-P.: Betriebsorganisation für Ingenieure, 6., aktualisierte Aufl., München/Wien 2008
94	<p>B2-Inf1.0 (01.10.2004 bis 30.09.2007)</p>

2.19 Recht

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Recht
Kürzel:	v56
Lehrveranstaltungen:	v560 Medienrecht v561 Datenschutz
Semester:	3 (v560), 6 (v561)
Modulverantwortliche(r):	Christian Buhl
Dozent(in):	Christian Buhl, Peter Münch
Sprache:	deutsch
Zuordnung zum Curriculum:	Informatik (Bachelor): Pflicht, 3. Sem., 6. Sem. Medieninformatik (Bachelor): Pflicht, 5. Sem., 6. Sem.
Lehrform / SWS:	Vorlesung: 5 SWS, Gruppengröße: 40
Arbeitsaufwand:	Präsenzstudium: 47 Stunden, Eigenstudium: 73 Stunden
Kreditpunkte:	4
Voraussetzungen:	
Lernziele / Kompetenzen:	<p>Vermitteln der Grundzüge des Medien- und des Urheberrechts; Entwicklung eines Problembewusstseins für juristische Fragestellungen aus diesen Rechtsgebieten; Studierende in die Lage versetzen, einfache juristische Probleme aus den vorgenannten Bereichen selbst zu lösen.</p> <p>Befähigung, im künftigen Einsatzgebiet die gesetzlichen und technisch-organisatorischen Anforderungen des Datenschutzes praktisch umzusetzen (datenschutzfreundliche Technik, Systemdatenschutz).</p> <p>Erlangung eines ersten Fachkundenachweises zur Befähigung, die Aufgabe eines Datenschutzbeauftragten wahrzunehmen.</p>

Inhalt:	<p>Medienrecht</p> <ul style="list-style-type: none"> • Einführung in das Medienrecht • Rechtliche Grundlagen des Medienrechts • Medienrechtliche Rechtsgrundsätze • Recht der Medienregulierung • Zivilrechtliche Ansprüche des Medienrechts • Strafrechtliche Vorschriften mit medienrechtlichem Bezug • Bestimmungen des Jugendschutzes • Grundzüge des Urheberrechts • Das Urheberrecht an Computerprogrammen, Datenbanken und Websites • Urheberrechtliche Aspekte verschiedener EDV-Verträge • Online-Recht <p>Datenschutz</p> <ul style="list-style-type: none"> • Gesetzliche Grundlagen des Datenschutzes <ul style="list-style-type: none"> – Wesentliche Grundlagen aus ausgewählten bereichsspezifischen und bereichsübergreifenden Datenschutzgesetzen – Rechte, Pflichten und Aufgabendurchführung des betrieblichen (behördlichen) Datenschutzbeauftragten • Technisch-organisatorischer Datenschutz <ul style="list-style-type: none"> – Risikomanagement und Basistechnologien zur Realisierung des technisch-organisatorischen Datenschutzes – Realisierung der gesetzlichen Anforderungen zum technisch-organisatorischen Datenschutz im Einzelnen – Auswahlverfahren zu geeigneten und angemessenen Sicherheitsmechanismen
Studien-/Prüfungsleistungen:	Klausur
Medienformen:	Beamerpräsentation

Literatur:	<p>Medienrecht</p> <ul style="list-style-type: none">• Fechner, Frank: Medienrecht, Verlag Mohr-Soebeck - UTB, Stuttgart, 5. Auflage, 2004• Paschke, Marian: Medienrecht, Springer-Verlag Berlin Heidelberg, 2. Auflage, 2001• Prinz, Matthias / Peters, Butz: Medienrecht, C. H. Beck, München, 1999• Reh binder, Manfred: Urheberrecht, C. H. Beck, München, 13. Auflage, 2004• Roßnagel, Alexander: Recht der Multimediadienste (Lsbl.), C. H. Beck, München <p>Datenschutz</p> <ul style="list-style-type: none">• Bundesdatenschutzgesetz (BDSG) vom Januar 2003• GDD: Datenschutz im Unternehmen• GDD: Datensicherheit im Unternehmen, Eigenverlag• Koch (Hrsg.): Handbuch des betrieblichen Datenschutzbeauftragten, Datakontext-Fachverlag• Münch: Technisch-organisatorischer Datenschutz, Datakontext-Fachverlag
------------	--

2.20 Projektmanagement

2.20.1 bis 30.09.2007 (Rb)

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Projektmanagement
Kürzel:	v60
Lehrveranstaltungen:	v600 Projektmanagement v601 Communication Skills v602 Lehrassistenz
Semester:	3 (v600, v601), 5 (v602)
Modulverantwortliche(r):	Ulrich Raubach
Dozent(in):	Dozenten, Hans Joachim Göttner, Ulrich Raubach
Sprache:	deutsch
Zuordnung zum Curriculum:	Informatik (Bachelor): Pflicht, 5. Sem.
Lehrform / SWS:	Vorlesung: 2 SWS, Gruppengröße: 125 Praktikum: 2 SWS, Gruppengröße: 10 - 20 Einzelleistung: 2 SWS
Arbeitsaufwand:	Präsenzstudium: 56 Stunden, Eigenstudium: 124 Stunden
Kreditpunkte:	6
Voraussetzungen:	
Lernziele / Kompetenzen:	<p>Projektmanagement:</p> <p>Kennen lernen und anwendungssichere Beherrschung der Verfahren zur Projektstrukturierung und -planung sowie des Projektcontrolling. Punktuelle Vertiefungen erfolgen im Rahmen der Bestimmung des Mengengerüsts (Zeit, Ressourcen) sowie der Konfliktbeherrschung in Engpassituationen sowie im Schnittstellenbereich aus Planungstechniken (Mengengerüst) und Kostengesichtspunkten (monetär bewertetes Mengengerüst), um der wachsenden Praxisrelevanz der kaufmännischen Funktionen innerhalb des Projektmanagement Ausdruck zu verleihen. Darüber hinaus sollen die Fähigkeiten zur konkreten Methodenauswahl und -modifikation in Abhängigkeit von der Projektkomplexität entwickelt werden; dies vor dem Hintergrund der Maxime „soviel wie nötig“.</p> <p>Communications Skills:</p> <p>Verbesserung der persönlichen soft skills für Studium/ Beruf. Sensibilisierung für menschliche Interaktionen und Betriebsprozesse.</p> <p>Üben eigener rhetorischer Fähigkeiten im Rahmen von Präsentationen, Vorträgen und Referaten. Vertiefung der sozialen Kompetenz; Erkennen der Bedeutung von verbalen und nonverbalen Signalen für die eigene Kommunikation.</p> <p>Bewerbungsübung und Interviewschulung Assessment Center - Training; Verhaltensregeln bei Teamarbeit/Projekten.</p> <p>Lehrassistenz:</p> <p>Die Studierenden sollen im Umgang und unter Anleitung der Dozenten mit Studierenden des 1. und 2. Semesters lernen, ihr Wissen und ihre Erfahrung in Informatik und Mathematik (z. B. in Programmiersprachen) im Rahmen von Übungen und Tutorien strukturiert weiterzugeben.</p>

Inhalt:	<p>Projektmanagement</p> <ul style="list-style-type: none"> • Begriffe, Definitionen, Abgrenzungen • Projektmanagement im Prozess der Unternehmensführung <ul style="list-style-type: none"> – Projektmanagement im System der Unternehmenspläne – Projektorganisation • Projektplanung <ul style="list-style-type: none"> – Grundprinzipien der Projektplanung – Methoden der Projektplanung • Projektcontrolling <ul style="list-style-type: none"> – Projektsteuerung – Risikomanagement in Projekten – Projektdokumentation und -berichtswesen <p>Communication Skills</p> <ul style="list-style-type: none"> • Anwendung des Kommunikationsmodell von Schulz von Thun <ul style="list-style-type: none"> – Üben situativer und personenbezogener Gesprächsführung – Konflikt-handhabung und Klärungsgespräche • Gruppenarbeit und Ergebnispräsentation <ul style="list-style-type: none"> – betriebliche Fallstudienbearbeitung – berufliche Meetings/Protokollführung – Verhaltenstraining bei Verkaufsgesprächen • Unternehmerische Entscheidungsfindung <ul style="list-style-type: none"> – praxisbezogene Postkorbübungen – Gesprächsführung mit Betriebsrat – Hinweise zur interkulturellen Kompetenz <p>Lehrassistenz</p> <ul style="list-style-type: none"> • Variiert in Abhängigkeit von der konkreten Aufgabenstellung. Inhalte orientieren sich am Vorlesungsinhalt. <p>Typische Tätigkeiten im Rahmen der Lehrassistenz:</p> <ul style="list-style-type: none"> • Unterstützung von Dozenten und Hochschulassistenten bei Übungsveranstaltungen für Kommilitonen/innen jüngerer Semester z. B. in den Rechenzentren • Entwurf von Übungsaufgaben und -materialien • Erstellung von Musterlösungen und die systematische Entwicklung von Testfällen für Übungsaufgaben • zusätzliche Betreuung/Beratung der Studierenden • Durchführung zusätzlicher Tutorien z. B. in Mathematik und Rechnungswesen • Aktualisierung von Veranstaltungsmaterialien (Handouts, Web-Seiten) • Zusammenstellung von Beispielen, Realisierung von Demo-Programmen • Quellenrecherche zu Veranstaltungsthemen
---------	---

	<p>Betreuung während der Lehrassistenz:</p> <ul style="list-style-type: none"> • Definition und Dokumentation einer Aufgabenstellung (Gegenstand und Umfang) durch den Dozenten • Regelmäßige Kommunikation mit dem Dozenten im Verlauf der Tätigkeit, abhängig von der Aufgabenstellung alle 1 - 2 Wochen • Statusberichte • (Zwischen-) Ergebnispräsentationen • Anleitung und Hinweise zur weiteren Tätigkeit • bei veranstaltungsbezogenen Tätigkeiten: Absprache der Inhalte der nächsten Veranstaltung • Abschlussbesprechung mit Dozenten • Abnahme Arbeitsergebnis • Diskussion des Verlaufs der Assistenz • Bewertung/Feedback durch den Dozenten
Studien-/Prüfungsleistungen:	Klausur, Testate
Medienformen:	Beamerpräsentation, Overheadfolien, Tafel, Rollenspiele
Literatur:	<p>Projektmanagement</p> <ul style="list-style-type: none"> • Burghardt, Manfred: Projektmanagement, 4. Auflage, Erlangen; München: Publicis-MCD-Verlag, 1997 • Diethelm, Gerd: Projektmanagement, Bd. 1: Grundlagen, Herne; Berlin: Verlag NWB, 2000 • Diethelm, Gerd: Projektmanagement, Bd. 2: Sonderfragen, Herne; Berlin: Verlag NWB, 2001 • Heinrich, Lutz J.: Management von Informatik-Projekten, München; Wien: Oldenbourg, 1997 • Leidig, Guido; Sommerfeld, Rita: Kalkulations- und Projekt-Management - Leitfaden für Digital- und Printmedien, Wiesbaden: Print & Media Form AG, 2003, Hrsg.: Bundesverband Druck und Medien e. V., Wiesbaden • Madauss, Bernd: Handbuch Projektmanagement, 5. überarbeitete und erweiterte Auflage, Stuttgart: Schäffer- Poeschel, 1994 • Michel, Reiner: Taschenbuch Projektcontrolling, Heidelberg: Sauer, 1993 • Schelle, Heinz: Projekte zum Erfolg führen, München: Verlag C. H. Beck, Reihe Wirtschaftsberater im dtv, 1996

- Schultz, Volker:
Projektkostenschätzung,
Wiesbaden: Gabler, 1995
- Schwarze, Jochen:
Netzplantechnik: eine Einführung in das Projektmanagement,
7. vollst. überarb. Auflage, Herne; Berlin: NWB-Studienbücher,
1994
- Wischniewski, Erik:
Modernes Projektmanagement,
4. vollst. überarb. und erw. Auflage, Braunschweig; Wiesbaden:
Vieweg, 1993

Communication Skills

- Jay, A.:
Die perfekte Präsentation,
Niederhausen 2002
- Argyle, M.:
Soziale Interaktion,
Köln 1998
- Golemann, D.:
Der Erfolgsquotient,
München 2000
- Kratz, H.-J.:
Chef-Checkliste Mitarbeiterführung,
Regensburg 1999
- Grüning, C.; Mielke, G.:
Präsentieren und Überzeugen. Das Kienbaum Trainingskonzept,
Freiburg 2003
- Staufenbiel, J.:
Berufsplanung für den IT-Nachwuchs,
Köln 2000
- Staufenbiel, J.:
Berufsplanung für Ingenieure,
Köln 2002
- Schulz von Thun, F.:
Miteinander Reden. Störungen und Klärungen, Teil 1 und 2,
Reinbek 2001
- Hesse/Schrader:
Neue Bewerbungstrategien für Hochschulabsolventen,
Frankfurt 2002

Lehrassistentenz

- Vorlesungsunterlagen und Themen sowie Umfang von Übungsaufgaben teilweise mit Lösungshinweisen werden von den Dozenten zur Verfügung gestellt.

2.20.2 ab 01.10.2007 (Stl)

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Projektmanagement
Kürzel:	v60
Lehrveranstaltungen:	v600 Projektmanagement v601 Communication Skills v602 Lehrassistenz
Semester:	3 (v600, v601), 5 (v602)
Modulverantwortliche(r):	Markus Stallkamp
Dozent(in):	Dozenten, Hans Joachim Göttner, Markus Stallkamp
Sprache:	deutsch
Zuordnung zum Curriculum:	Informatik (Bachelor): Pflicht, 5. Sem.
Lehrform / SWS:	Vorlesung: 2 SWS, Gruppengröße: 125 Praktikum: 2 SWS, Gruppengröße: 10 - 20 Einzelleistung: 2 SWS
Arbeitsaufwand:	Präsenzstudium: 56 Stunden, Eigenstudium: 124 Stunden
Kreditpunkte:	6
Voraussetzungen:	
Lernziele / Kompetenzen:	<p>Projektmanagement:</p> <p>Kennen lernen der grundlegenden Begriffe und Techniken entsprechend den Phasen eines Projekts;</p> <p>Vermittlung grundlegender Kommunikationsfähigkeiten (präsentieren, diskutieren, moderieren und verhandeln).</p> <p>Communications Skills:</p> <p>Verbesserung der persönlichen soft skills für Studium/ Beruf. Sensibilisierung für menschliche Interaktionen und Betriebsprozesse.</p> <p>Üben eigener rhetorischer Fähigkeiten im Rahmen von Präsentationen, Vorträgen und Referaten. Vertiefung der sozialen Kompetenz; Erkennen der Bedeutung von verbalen und nonverbalen Signalen für die eigene Kommunikation.</p> <p>Bewerbungsübung und Interviewschulung Assessment Center - Training; Verhaltensregeln bei Teamarbeit/Projekten.</p> <p>Lehrassistenz:</p> <p>Die Studierenden sollen im Umgang und unter Anleitung der Dozenten mit Studierenden des 1. und 2. Semesters lernen, ihr Wissen und ihre Erfahrung in Informatik und Mathematik (z. B. in Programmiersprachen) im Rahmen von Übungen und Tutorien strukturiert weiterzugeben.</p>

Inhalt:	<p>Projektmanagement</p> <ul style="list-style-type: none"> • Nach einer kurzen Einführung wird Phase für Phase eines typischen Projekts vorgestellt. Für jede Phase werden dezidiert Begriffe, Aktionen sowie Techniken vorgestellt. Die Phasen sind: <ul style="list-style-type: none"> – Projektdefinition mit Projektantrag – Projektplanung mit Projektplan – Projektkontrolle mit Projektbericht – Projektabschluss mit Abschlussbericht • Abschließend werden noch Sonderthemen des Projektmanagements präsentiert. Hierzu zählen beispielsweise aktuelle Projektbeispiele und typische Stolpersteine des Projektmanagements. <p>Communication Skills</p> <ul style="list-style-type: none"> • Anwendung des Kommunikationsmodell von Schulz von Thun <ul style="list-style-type: none"> – Üben situativer und personenbezogener Gesprächsführung – Konflikt-handhabung und Klärungsgespräche • Gruppenarbeit und Ergebnispräsentation <ul style="list-style-type: none"> – betriebliche Fallstudienbearbeitung – berufliche Meetings/Protokollführung – Verhaltenstraining bei Verkaufsgesprächen • Unternehmerische Entscheidungsfindung <ul style="list-style-type: none"> – praxisbezogene Postkorbübungen – Gesprächsführung mit Betriebsrat – Hinweise zur interkulturellen Kompetenz <p>Lehrassistenz</p> <ul style="list-style-type: none"> • Variiert in Abhängigkeit von der konkreten Aufgabenstellung. Inhalte orientieren sich am Vorlesungsinhalt. <p>Typische Tätigkeiten im Rahmen der Lehrassistenz:</p> <ul style="list-style-type: none"> • Unterstützung von Dozenten und Hochschulassistenten bei Übungsveranstaltungen für Kommilitonen/innen jüngerer Semester z. B. in den Rechenzentren • Entwurf von Übungsaufgaben und -materialien • Erstellung von Musterlösungen und die systematische Entwicklung von Testfällen für Übungsaufgaben • zusätzliche Betreuung/Beratung der Studierenden • Durchführung zusätzlicher Tutorien z. B. in Mathematik und Rechnungswesen • Aktualisierung von Veranstaltungsmaterialien (Handouts, Web-Seiten) • Zusammenstellung von Beispielen, Realisierung von Demo-Programmen • Quellenrecherche zu Veranstaltungsthemen
---------	--

	<p>Betreuung während der Lehrassistenz:</p> <ul style="list-style-type: none"> • Definition und Dokumentation einer Aufgabenstellung (Gegenstand und Umfang) durch den Dozenten • Regelmäßige Kommunikation mit dem Dozenten im Verlauf der Tätigkeit, abhängig von der Aufgabenstellung alle 1 - 2 Wochen • Statusberichte • (Zwischen-) Ergebnispräsentationen • Anleitung und Hinweise zur weiteren Tätigkeit • bei veranstaltungsbezogenen Tätigkeiten: Absprache der Inhalte der nächsten Veranstaltung • Abschlussbesprechung mit Dozenten • Abnahme Arbeitsergebnis • Diskussion des Verlaufs der Assistenz • Bewertung/Feedback durch den Dozenten
Studien-/Prüfungsleistungen:	Klausur, Testate
Medienformen:	Beamerpräsentation, Overheadfolien, Tafel, Rollenspiele

Literatur:	<p>Projektmanagement</p> <ul style="list-style-type: none"> • Burghardt, Manfred: Einführung in Projektmanagement, 7. Auflage, Publicis Corporate Publishing, Erlangen, 2006 • Burghardt, Manfred: Projektmanagement - Leitfaden für die Planung, Überwachung und Steuerung von Projekten, 5. Auflage, Publicis Corporate Publishing, Erlangen, 2007 • DeMarco, Tom: Der Termin - Ein Roman über Projektmanagement, Carl Hanser Verlag, München, 1998 • Tumuscheit, Klaus D.: Überleben im Projekt - 10 Projektfallen und wie man sie um- geht, Orell Füssli Verlag, Zürich, 2007 <p>Communication Skills</p> <ul style="list-style-type: none"> • Jay, A.: Die perfekte Präsentation, Niederhausen 2002 • Argyle, M.: Soziale Interaktion, Köln 1998 • Golemann, D.: Der Erfolgsquotient, München 2000 • Kratz, H.-J.: Chef-Checkliste Mitarbeiterführung, Regensburg 1999 • Grüning, C.; Mielke, G.: Präsentieren und Überzeugen. Das Kienbaum Trainingskon- zept, Freiburg 2003 • Staufenbiel, J.: Berufsplanung für den IT-Nachwuchs, Köln 2000 • Staufenbiel, J.: Berufsplanung für Ingenieure, Köln 2002 • Schulz von Thun, F.: Miteinander Reden. Störungen und Klärungen, Teil 1 und 2, Reinbek 2001 • Hesse/Schrader: Neue Bewerbungstrategien für Hochschulabsolventen, Frankfurt 2002 <p>Lehrassistenz</p> <ul style="list-style-type: none"> • Vorlesungsunterlagen und Themen sowie Umfang von Übungs- aufgaben teilweise mit Lösungshinweisen werden von den Do- zenten zur Verfügung gestellt.
------------	---

2.21 Seminar

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Seminar
Kürzel:	v80
Lehrveranstaltungen:	v800 Seminar Informatik
Semester:	5
Modulverantwortliche(r):	Uwe Schmidt
Dozent(in):	Dozenten
Sprache:	deutsch
Zuordnung zum Curriculum:	Informatik (Bachelor): Pflicht, 5. Sem.
Lehrform / SWS:	Seminar: 2 SWS, Gruppengröße: 12 - 15
Arbeitsaufwand:	Präsenzstudium: 20 Stunden, Eigenstudium: 100 Stunden
Kreditpunkte:	4
Voraussetzungen:	
Lernziele / Kompetenzen:	<p>Eigenständiges Einarbeiten in ein anspruchsvolles Thema über aktuelle Entwicklungen in der angewandten Informatik;</p> <p>Gezielte Literaturrecherche, insbesondere unter Berücksichtigung der Quellen des Internet;</p> <p>Optimierung der Fähigkeit des freien Vortrags, des Umgangs mit Präsentationsmedien und der offenen Diskussion wissenschaftlicher Themen in der Gruppe;</p> <p>Erstellung einer stilistisch und fachlich ansprechenden Ausarbeitung als Vorbereitung für die Bachelor-Arbeit.</p>
Inhalt:	<ul style="list-style-type: none"> • nach Aufgabenstellung unterschiedlich • 12 - 15 Einzelvorträge von Seminarteilnehmern pro Semester • Themenbeispiele: http://www.fh-wedel.de/~si/seminare/index.html
Studien-/Prüfungsleistungen:	bewertete Fachvorträge, bewerteter Abschlussbericht zum jeweiligen Einzelthema
Medienformen:	Beamerpräsentation, Softwaredemonstration, Tafel
Literatur:	Recherche nach Aufgabenstellung

2.22 Bachelor-Thesis

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Bachelor-Thesis
Kürzel:	v98
Lehrveranstaltungen:	v980 Betriebspraktikum v999 Bachelor-Thesis
Semester:	6
Modulverantwortliche(r):	Uwe Schmidt
Dozent(in):	Dozenten
Sprache:	deutsch
Zuordnung zum Curriculum:	Informatik (Bachelor): Pflicht, 6. Sem.
Lehrform / SWS:	Einzelbetreuung, 0 SWS, Gruppengröße: 1 - 2
Arbeitsaufwand:	Präsenzstudium: 2 Stunden, Eigenstudium: 838 Stunden
Kreditpunkte:	28 (Betriebspraktikum: 16, Bachelor-Thesis: 12)
Voraussetzungen:	
Lernziele / Kompetenzen:	<p>Erfahrungen sammeln für die spätere berufliche Tätigkeit in einem Unternehmen.</p> <p>Soziale Kompetenzen: Selbständigkeit, Eigenverantwortliches Handeln, Kommunikation und Teamfähigkeit, Zeitmanagement;</p> <p>Fachliche Kompetenzen: Methodisches Arbeiten, praktisches Anwenden der im Studium erlernten Kenntnisse.</p>
Inhalt:	<p>Betriebspraktikum</p> <ul style="list-style-type: none"> Im Rahmen des Betriebspraktikums sollen sich die Studierenden in die Inhalte und Abläufe der Abteilung oder des Bereiches in einem Unternehmen einarbeiten und das dortige Tagesgeschäft unterstützen. Gleichzeitig kann in Abstimmung mit je einem Mitarbeiter des Unternehmens und einem Dozenten der Hochschule das Aufgabenfeld inhaltlich umrissen werden, in dem im Anschluss an das Praktikum eine Bachelor-Thesis erarbeitet werden kann. <p>Bachelor-Thesis</p> <ul style="list-style-type: none"> Die Bachelor-Thesis soll im Regelfall in Kooperation mit einem Unternehmen erarbeitet werden. Themen aus den Arbeitsgruppen und Laboren der Hochschule sind ebenfalls möglich. Die Arbeit ist als abschließende, vom Studierenden eigenständig aber hochschul- und unternehmensseitig betreutes Projekt zu verstehen. Im Sinne der Zielsetzung der Bachelor-Ausbildung, der Erlangung des ersten berufsqualifizierenden Abschlusses, ist die Arbeit thematisch an einer Problemstellung eines kooperierenden Unternehmens orientiert oder sie besteht aus einer praxisrelevanten hochschulinternen Aufgabe. Dabei kann es sich um rein informatische aber auch um Inhalte aus Anwendungsgebieten handeln, deren Lösung überwiegend Kenntnisse aus der Informatik erfordert. Die Studierenden sollen mit ihrer Arbeit den Nachweis erbringen, dass sie in der Lage sind, auf wissenschaftlicher Basis und eigenständig eine Problemlösung zu erarbeiten. Wesentlich sind strukturierte und argumentierte Inhalte sowie das Einhalten üblicher Formalia. Genauere Angaben sind der Homepage des jeweils betreuenden Dozenten zu entnehmen.
Studien-/Prüfungsleistungen:	Praktikumsnachweis, schriftliche Arbeit
Medienformen:	themenabhängig
Literatur:	themenabhängig