

**Modulhandbuch**  
**Studiengang Bachelor of Science Softwaretechnik**  
**Prüfungsordnung: 2012**

Wintersemester 2015/16  
Stand: 08. April 2016

Universität Stuttgart  
Keplerstr. 7  
70174 Stuttgart

## Kontaktpersonen:

---

Studiendekan/in:	Univ.-Prof. Stefan Wagner Institut für Softwaretechnologie Tel.: 0711/685-88455 E-Mail: stefan.wagner@informatik.uni-stuttgart.de
Studiengangsmanger/in:	Katrin Schneider Institut für Maschinelle Sprachverarbeitung Tel.: 685 88520 E-Mail: katrin.schneider@informatik.uni-stuttgart.de
Prüfungsausschussvorsitzende/r:	Univ.-Prof. Lars Grunske Zuverlässige Softwaresysteme Tel.: E-Mail: lars.grunske@iste.uni-stuttgart.de
Fachstudienberater/in:	Katrin Schneider Institut für Maschinelle Sprachverarbeitung Tel.: 685 88520 E-Mail: katrin.schneider@informatik.uni-stuttgart.de
Stundenplanverantwortliche/r:	Apl. Prof. Ulrich Hertrampf Institut für Formale Methoden der Informatik Tel.: 7816-344 E-Mail: ulrich.hertrampf@f05.uni-stuttgart.de

# Inhaltsverzeichnis

<b>Präambel</b> .....	<b>6</b>
<b>Qualifikationsziele</b> .....	<b>7</b>
<b>100 Basismodule</b> .....	<b>8</b>
12060 Datenstrukturen und Algorithmen .....	9
17210 Einführung in die Softwaretechnik .....	11
10190 Mathematik für Informatiker und Softwaretechniker .....	13
10280 Programmierung und Software-Entwicklung .....	15
16520 Software-Qualität .....	17
10940 Theoretische Grundlagen der Informatik .....	18
<b>200 Kernmodule</b> .....	<b>20</b>
11890 Algorithmen und Berechenbarkeit .....	21
14360 Einführung in die Technische Informatik .....	22
14370 Fachstudie Softwaretechnik .....	24
14390 Programmentwicklung .....	25
36100 Programmierparadigmen .....	26
42790 Seminar SWT .....	28
14480 Sichere und zuverlässige Softwaresysteme .....	30
16500 Software Engineering .....	31
16510 Software-Praktikum .....	32
41940 Studienprojekt-Th .....	33
<b>300 Ergänzungsmodule</b> .....	<b>34</b>
310 Katalog SWT .....	35
41590 Einführung in die Numerik und Stochastik für Softwaretechniker .....	36
10210 Mensch-Computer-Interaktion .....	38
10220 Modellierung .....	40
36530 Rechnerorganisation 1 .....	42
40090 Systemkonzepte und -programmierung .....	44
320 Katalog ISG .....	46
10030 Architektur von Anwendungssystemen .....	47
10060 Computergraphik .....	49
41590 Einführung in die Numerik und Stochastik für Softwaretechniker .....	51
18560 Grundlagen der Eingebetteten Systeme .....	53
10100 Grundlagen der Graphischen Ingenieursysteme .....	54
10110 Grundlagen der Künstlichen Intelligenz .....	56
56930 Grundlagen der Rechnerarchitektur .....	58
42410 Grundlagen des Wissenschaftlichen Rechnens .....	60
10170 Imaging Science .....	62
10180 Information Retrieval und Text Mining .....	64
10210 Mensch-Computer-Interaktion .....	65
10220 Modellierung .....	67
39040 Rechnernetze .....	69
36530 Rechnerorganisation 1 .....	71
40090 Systemkonzepte und -programmierung .....	73
330 Katalog ISW .....	75
10030 Architektur von Anwendungssystemen .....	76
10060 Computergraphik .....	78
41590 Einführung in die Numerik und Stochastik für Softwaretechniker .....	80

18560 Grundlagen der Eingebetteten Systeme .....	82
10100 Grundlagen der Graphischen Ingenieursysteme .....	83
10110 Grundlagen der Künstlichen Intelligenz .....	85
56930 Grundlagen der Rechnerarchitektur .....	87
42410 Grundlagen des Wissenschaftlichen Rechnens .....	89
10170 Imaging Science .....	91
10180 Information Retrieval und Text Mining .....	93
10210 Mensch-Computer-Interaktion .....	94
10220 Modellierung .....	96
39040 Rechnernetze .....	98
36530 Rechnerorganisation 1 .....	100
40090 Systemkonzepte und -programmierung .....	102
340 Wahlmodule aus Master SWT .....	104
60860 3D Scanner - Algorithms and Systems .....	106
42910 Advanced Business Process Management .....	107
55600 Advanced Information Management .....	108
55740 Advanced Service Computing .....	109
29550 Algorithmische Geometrie .....	111
29760 Algorithmische Gruppentheorie .....	112
45760 Ausgewählte Kapitel der Algorithmentheorie .....	114
42480 Ausgewählte Kapitel des Wissenschaftlichen Rechnens .....	115
56680 Automaten über unendlichen Objekten .....	116
10040 Bildsynthese .....	118
42900 Business Process Management .....	120
57050 Compilerbau .....	122
29570 Computer Interface Technologien .....	124
29430 Computer Vision .....	125
55640 Correspondence Problems in Computer Vision .....	127
29580 Data Compression .....	129
48480 Data Engineering .....	130
55620 Data Warehousing, Data Mining, and OLAP .....	132
10080 Datenbanken und Informationssysteme .....	134
29600 Digital System Design II .....	136
29590 Digitale Systeme .....	137
39250 Distributed Systems I .....	138
45730 Distributed Systems II .....	140
29710 Embedded Systems Engineering .....	142
58190 Entwurf und Implementierung eines Compilers .....	143
56980 Fachpraktikum Cloud Architekturen und Management .....	144
29740 Fachpraktikum Eingebettete Systeme .....	145
29750 Fachpraktikum Rechnerarchitektur .....	146
24900 Fachpraktikum Rechnergestützte Methoden der Produktentwicklung .....	147
45750 Fachpraktikum Verteilte Systeme .....	149
58440 Fachpraktikum: Algorithmen .....	150
29620 Fortgeschrittene Aspekte der Sprachperzeption und Sprachproduktion .....	151
29440 Geometric Modeling and Computer Animation .....	152
29450 Graphentheorie .....	154
29610 Hardware Based Fault Tolerance .....	156
14380 Hardware Verification and Quality Assessment .....	158
42920 Hardware-Software-Codesign .....	160
42860 Hauptseminar (Master SWT 1) .....	161
55560 Hauptseminar (Master SWT 2) .....	163
42420 High Performance Computing .....	165
51720 IT-Strategy .....	167
48500 Image Synthesis .....	169
55610 Information Integration .....	171
55630 Information Visualization and Visual Analytics .....	173
60120 Interaktive Systeme .....	175

29460 Kryptographische Verfahren .....	176
29480 Loose Coupling and Message Based Applications .....	177
29470 Machine Learning .....	179
29640 Mikrocontroller .....	181
29720 Mobile Computing .....	183
10120 Modellbildung und Simulation .....	185
29730 Modelling, Simulation, and Specification .....	187
55650 Multimodal Interaction for Ubiquitous Computers .....	189
42460 Numerische Simulation .....	190
40680 Optimization .....	191
56790 Parallele Numerik .....	193
29650 Parallele Programmierung .....	195
10250 Parallele Systeme .....	196
48550 Practical Course Information Systems .....	197
48570 Practical Course Visual Computing .....	198
29660 Programmanalysen und Compilerbau .....	199
51740 Quantencomputing .....	201
29670 Rapid Prototyping .....	202
29680 Real-Time Programming .....	203
29690 Real-Time Video Processing I .....	204
29700 Real-Time Video Processing II .....	205
45740 Rechnernetze II .....	206
48580 Reinforcement Learning .....	208
48600 Robotics I .....	210
48620 Scientific Visualization .....	211
29510 Service Computing .....	213
46660 Service Management and Cloud Computing, and Evaluation .....	215
42520 Services and Service Composition .....	216
56550 Software Verification .....	218
42840 Software-Recht .....	220
60140 Sprachbau mit Language Workbenches .....	221
29500 Visual Computing .....	223
<b>400 Schlüsselqualifikationen fachaffin .....</b>	<b>225</b>
16610 Studienprojekt-Pr .....	226
<b>81110 Bachelorarbeit Softwaretechnik .....</b>	<b>228</b>

## Präambel

Wie andere Ingenieure auch, arbeiten die meisten Informatikerinnen und Informatiker nach ihrer Ausbildung konstruktiv. Sie entwickeln also neue oder verändern, verbessern und erweitern bestehende Software. Daran ist der Studiengang *Softwaretechnik* ausgerichtet: Er betont die konstruktiven Aspekte der Informatik. Der Lehrstoff, der den Studierenden vermittelt wird, überlappt mit dem des Studiengangs Informatik, aber in der *Softwaretechnik* ist es wichtig, dass das Gehörte auch angewendet wird. Die Absolventen der *Softwaretechnik* sind damit besonders gut auf alle Informatikberufe vorbereitet, in denen an Software gearbeitet wird, auch auf die Managementaufgaben, die die meisten früher oder später übernehmen werden.

In jedem Softwareprojekt wird das Ziel verfolgt, Software so zu entwickeln oder zu verändern, dass sie am Ende den Anforderungen der Kunden hinsichtlich Funktion und Qualität entspricht. Dabei sollen die Kosten und die Entwicklungszeit möglichst niedrig sein. Dieses Ziel wird den Studierenden der *Softwaretechnik* in Vorlesungen und Übungen, vor allem aber in einer Reihe von Projekten, vermittelt bis hin zum zwölfmonatigen Studienprojekt, in dem etwa zehn Studierende gemeinsam und selbstorganisiert eine anspruchsvolle Entwicklungsaufgabe lösen. Neben der Technik werden damit Arbeit im Team, Kommunikation und Präsentation erlernt und geübt.

Auf den Bachelor-Studiengang *Softwaretechnik* baut der gleichnamige Master-Studiengang auf. Den *Softwaretechnik*-Absolventen stehen aber auch die anderen Masterstudiengänge der Informatik offen.

## Qualifikationsziele

Das Curriculum des Studienganges sieht im 1. und 2. Semester eine Vertiefung der mathematischen und technischen Grundlagen der Informatik und Softwareentwicklung in Pflichtmodulen vor.

Im 3. bis 6. Semester liegt der Schwerpunkt der Ausbildung auf diversen Grundlagen- und Spezialisierungsfächern verschiedener Bereiche der Informatik und Softwaretechnik. Zusätzliche Inhalte sind Seminare, fachliche Module als Wahlpflichtbereich sowie fachübergreifende Schlüsselqualifikationen. Die praktische Ausbildung wird durch ein Studienprojekt vertieft, in dem ein komplexes Softwareprojekt im Team von ca. 10 Personen innerhalb von 12 Monaten durchgeführt wird. Im 6. Semester liegt ein weiterer Schwerpunkt auf der Bachelorarbeit. Mit der Bachelorarbeit ist die Befähigung zu zeigen, innerhalb einer vorgegebenen Frist eine komplexe Aufgabenstellung aus dem Bereich der Softwaretechnik selbständig nach wissenschaftlichen Methoden zu bearbeiten und die Ergebnisse sachgerecht darzustellen. Die Absolventinnen und Absolventen des Bachelorstudienganges Softwaretechnik

- verfügen über ein vertieftes mathematisches und ingenieurwissenschaftliches Wissen, welches sie befähigt, Probleme und Aufgabenstellungen der Softwareentwicklung zu verstehen, kritisch einzuschätzen und zu lösen.
- sind in der Lage, jenes erlangte Wissen auf Erkenntnisse der Ingenieurwissenschaften anzuwenden.
- verfügen über Fachwissen auf Gebieten der Softwaretechnik und Informatik und können Aufgaben u.a. der Softwareentwicklung wissenschaftlich erkennen, beschreiben und bewerten, analysieren und lösen.
- haben umfassendes Verständnis über Softwareentwicklungsmethoden, ihre Anwendungsmöglichkeiten und verfügen über die Fertigkeit, Konzepte und Lösungen für neue Softwaresysteme zu entwickeln.
- verfügen über eine verantwortliche und selbständige wissenschaftliche Arbeitsweise. Die Beschäftigungsbereiche der Absolventinnen und Absolventen liegen u. a. in der Datenverarbeitungs- und Entwicklungsindustrie und in industriellen, kommerziellen, verwaltungstechnischen und wissenschaftlichen Anwendungen von Datenverarbeitungssystemen.

## 100 Basismodule

---

Zugeordnete Module:	10190	Mathematik für Informatiker und Softwaretechniker
	10280	Programmierung und Software-Entwicklung
	10940	Theoretische Grundlagen der Informatik
	12060	Datenstrukturen und Algorithmen
	16520	Software-Qualität
	17210	Einführung in die Softwaretechnik

---

## Modul: 12060 Datenstrukturen und Algorithmen

2. Modulkürzel:	051510005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	9.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	6.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Daniel Weiskopf		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Andrés Bruhn</li> <li>• Thomas Ertl</li> <li>• Stefan Funke</li> <li>• Daniel Weiskopf</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009, 2. Semester → Basismodule  B.Sc. Softwaretechnik, PO 2012, 2. Semester → Basismodule		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 10280 Programmierung und Software-Entwicklung</li> </ul>		
12. Lernziele:	<p>Die Studierenden kennen nach engagierter Mitarbeit in dieser Veranstaltung diverse zentrale Algorithmen auf geeigneten Datenstrukturen, die für eine effiziente Nutzung von Computern unverzichtbar sind. Sie können am Ende zu gängigen Problemen geeignete programmiersprachliche Lösungen angeben und diese in einer konkreten Programmiersprache formulieren.</p> <p>Die Lernziele lassen sich wie folgt zusammenfassen:</p> <ul style="list-style-type: none"> <li>• Kenntnis der Eigenschaften elementarer und häufig benötigter Algorithmen</li> <li>• Verständnis für die Auswirkungen theoretischer und tatsächlicher Komplexität</li> <li>• Erweiterung der Kompetenz im Entwurf und Verstehen von Algorithmen und der zugehörigen Datenstrukturen</li> <li>• Erste Begegnung mit nebenläufigen Algorithmen</li> </ul>		
13. Inhalt:	<p>Es werden die folgenden Themen behandelt:</p> <ul style="list-style-type: none"> <li>• Vorgehensweise bei der Entwicklung und Implementierung von Algorithmen</li> <li>• Komplexität und Effizienz von Algorithmen, O-Notation</li> <li>• Listen (Stack, Queue, doppelt verkettete Listen)</li> <li>• Sortierverfahren (Selection-, Insertion-, Bubble-, Merge-, Quick-Sort)</li> <li>• Bäume (Binär-, AVL-, 2-3-4-, Rot-Schwarz-, B-Bäume, Suchbäume, Traversierung, Heap)</li> <li>• Räumliche Datenstrukturen (uniforme Gitter, Oktal-, BSP-, kD-, CSG-Bäume, Bounding-Volumes)</li> <li>• Graphen (Datenstrukturen, DFS, BFS, topologische Traversierung, Dijkstra-, A*-, Bellman-Ford-Algorithmen, minimale Spannbäume, maximaler Fluss)</li> <li>• Räumliche Graphen (Triangulierung, Voronoi, Delaunay, Graph-Layout)</li> <li>• Textalgorithmen (String-Matching, Knuth-Morris-Pratt, Boyer-Moore, reguläre Ausdrücke, Levenshtein-Distanz)</li> <li>• Hashing (Hashfunktionen, Kollisionen)</li> <li>• Verteilte Algorithmen (Petri-Netze, Programmieren nebenläufiger Abläufe, einige parallele und parallelisierte Algorithmen)</li> </ul>		

- Algorithmenentwurf und -muster (inkrementell, greedy, divide-and-conquer, dynamische Programmierung, Backtracking, randomisierte Algorithmen)
- Maschinelles Lernen (überwachtes Lernen, Entscheidungsbäume, SVM, neuronale Netze; unüberwachtes Lernen, k-Means)

---

14. Literatur:

- G. Saake, K. Sattler. *Algorithmen und Datenstrukturen: Eine Einführung mit Java* . 5. Auflage, dpunkt-Verlag, 2013
- T. Ottmann, P. Widmayer. *Algorithmen und Datenstrukturen* . 5. Auflage, Springer-Verlag, 2012

---

15. Lehrveranstaltungen und -formen:

- 120601 Vorlesung Datenstrukturen und Algorithmen
- 120602 Übung Datenstrukturen und Algorithmen

---

16. Abschätzung Arbeitsaufwand:

Präsenzzeit:	63 h
Selbststudiums- /	207
Nachbearbeitungszeit:	
Summe:	270 h

---

17. Prüfungsnummer/n und -name:

- 12061 Datenstrukturen und Algorithmen (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein. Die genauen Details der Übungsleistungen und Ihrer Anrechnung werden zu Beginn der Lehrveranstaltung bekannt gegeben.
- V Vorleistung (USL-V), schriftlich, eventuell mündlich

---

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von:

Institut für Visualisierung und Interaktive Systeme

---

## Modul: 17210 Einführung in die Softwaretechnik

2. Modulkürzel:	051520015	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Stefan Wagner		
9. Dozenten:	Stefan Wagner		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009, 2. Semester → Basismodule B.Sc. Softwaretechnik, PO 2012, 2. Semester → Basismodule		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 10280 Programmierung und Software-Entwicklung</li> <li>• Modul 12060 Datenstrukturen und Algorithmen</li> <li>• Programmiererfahrung</li> </ul>		
12. Lernziele:	Die Veranstaltung liefert einen ersten Einblick in die Softwaretechnik. Sie ist abgestimmt auf die Software-Qualität im 1. und Programmentwicklung im 3. Semester.  Die Teilnehmer kennen die Grundbegriffe der Softwaretechnik und haben wichtige Techniken des Softwareprojekt-Managements und der Software-Entwicklung erlernt. Sie kennen Scrum als eine konkrete Vorgehensweise zur Softwareentwicklung		
13. Inhalt:	Die Vorlesung behandelt technische und andere Aspekte der Softwarebearbeitung, wie sie in der Praxis stattfindet. Die einzelnen Themen sind: <ul style="list-style-type: none"> <li>• Abgrenzung und Motivation des Software Engineerings</li> <li>• Vorgehensmodelle, agiles Vorgehen, Scrum</li> <li>• Software-Management</li> <li>• Software-Prüfung und Qualitätssicherung</li> <li>• Methoden, Sprachen und Werkzeuge für die einzelnen Phasen: Spezifikation, Grobentwurf, Feinentwurf, Implementierung, Test</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Ludewig, Licher: Software Engineering. dpunkt-Verlag, Heidelberg. 2. Aufl. 2010</li> <li>• Pfleeger, Atlee: Software Engineering. Pearson, 2010</li> <li>• Rubin: Essential Scrum. Addison-Wesley, 2013</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 172101 Vorlesung Einführung in die Softwaretechnik</li> <li>• 172102 Übung Einführung in die Softwaretechnik</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 17211 Einführung in die Softwaretechnik (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0,</li> <li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich, Hausaufgaben</li> </ul>		
18. Grundlage für ... :	<ul style="list-style-type: none"> <li>• 16500 Software Engineering</li> <li>• 16510 Software-Praktikum</li> </ul>		

19. Medienform: 

- Folien am Beamer unterstützt durch Tafel und Overhead
- Dokumente, Links und Diskussionsforum in ILIAS

---

20. Angeboten von: Software-Engineering

---

**Modul: 10190 Mathematik für Informatiker und Softwaretechniker**

2. Modulkürzel:	080300100	5. Moduldauer:	2 Semester
3. Leistungspunkte:	18.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	12.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	PD Andreas Markus Kollross		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Peter Lesky</li> <li>• Wolfgang Rump</li> <li>• Wolf-Patrick Düll</li> <li>• Andreas Markus Kollross</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009, 1. Semester → Basismodule  B.Sc. Softwaretechnik, PO 2012, 1. Semester → Basismodule		
11. Empfohlene Voraussetzungen:	Keine, die Teilnahme an einem Mathematik Vorkurs wird empfohlen.		
12. Lernziele:	Die Studierenden haben die mathematischen Grundlagen für die Studiengänge Informatik bzw. Softwaretechnik erarbeitet und den selbständigen und kreativen Umgang mit den mathematischen Stoffgebieten gelernt.		
13. Inhalt:	1. Semester: <ul style="list-style-type: none"> <li>• Grundlagen (Aussagenlogik, Mengen, Relationen, Abbildungen, Zahlenmengen, Grundbegriffe der Algebra)</li> <li>• Lineare Algebra (Vektorräume, lineare Abbildungen, Matrizen, Determinanten, lineare Gleichungssysteme, Eigenwerte, Normalformen, Hauptachsentransformation, Skalarprodukte)</li> <li>• Analysis (Konvergenz, Zahlenfolgen und Zahlenreihen, stetige Abbildungen, Folgen und Reihen von Funktionen, spezielle Funktionen)</li> </ul> 2. Semester: <ul style="list-style-type: none"> <li>• Differential- und Integralrechnung (Funktionen einer und mehrerer Variablen, Ableitungen, Taylorentwicklungen, Extremwerte, Integration, Anwendungen)</li> <li>• Gewöhnliche Differentialgleichungen (elementar lösbare Differentialgleichungen, Existenz und Eindeutigkeit von Lösungen)</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Anna Sändig, Mathematik, Vorlesungskripte , SS 2007</li> <li>• D. Hachenberger, Mathematik für Informatiker, 2005</li> <li>• M. Brill, Mathematik für Informatiker, 2001</li> <li>• P.Hartmann, Mathematik für Informatiker, 2002</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 101901 Vorlesung Mathematik 1 für Informatik und Softwaretechnik</li> <li>• 101902 Übung Mathematik 1 für Informatik und Softwaretechnik</li> <li>• 101903 Vorlesung Mathematik 2 für Informatik und Softwaretechnik</li> <li>• 101904 Übung Mathematik 2 für Informatik und Softwaretechnik</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 126 Stunden Nachbearbeitungszeit: 414 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 10191 Mathematik für Informatiker und Softwaretechniker (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Ein</li> </ul>		

- V Übungsschein aus den beiden Veranstaltungen, jeweils im 1. oder 2. Fachsemester zu erwerben  
Vorleistung (USL-V), schriftlich, eventuell mündlich
- 

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von:

---

## Modul: 10280 Programmierung und Software-Entwicklung

2. Modulkürzel:	051520005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	9.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	6.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	Frank Leymann		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009, 1. Semester → Basismodule B.Sc. Softwaretechnik, PO 2012, 1. Semester → Basismodule		
11. Empfohlene Voraussetzungen:	Keine. Teilnahme an einem Vorkurs Java ist hilfreich aber nicht notwendig.		
12. Lernziele:	Die Teilnehmer haben einen Überblick über das Gebiet der Informatik. Sie haben die wichtigsten Konzepte einer höheren Programmiersprache und ihrer Verwendung verstanden und sind in der Lage, kleine Programme (bis zu einigen hundert Zeilen) zu analysieren und selbst zu konzipieren und zu implementieren. Sie kennen die Möglichkeiten, Daten- und Ablaufstrukturen zu entwerfen, zu beschreiben und zu codieren. Sie haben die Abstraktionskonzepte moderner Programmiersprachen verstanden. Sie kennen die Techniken und Notationen zur Definition kontextfreier Programmiersprachen und können damit arbeiten.		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Die Programmiersprache Java und die virtuelle Maschine</li> <li>• Objekte, Klassen, Schnittstellen, Blöcke, Programmstrukturen, Kontrakte</li> <li>• Klassenmodellierung mit der UML</li> <li>• Objekterzeugung und -ausführung</li> <li>• Boolesche Logik</li> <li>• Verzweigungen, Schleifen, Routinen, Abstraktionen, Modularisierung, Variablen, Zuweisungen</li> <li>• Rechner, Hardware</li> <li>• Syntaxdarstellungen</li> <li>• Übersicht über Programmiersprachen und -werkzeuge</li> <li>• Grundlegende Datenstrukturen und Algorithmen</li> <li>• Vererbung, Polymorphe</li> <li>• Semantik</li> <li>• Programmierung graphischer Oberflächen</li> <li>• Übergang zum Software Engineering</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Appelrath, Hans-Jürgen und Ludewig, Jochen, "Skriptum Informatik - eine konventionelle Einführung", Verlag der Fachvereine Zürich und B.G. Teubner Stuttgart, 4. Auflage 1999</li> <li>• Meyer, Bertrand, "Touch of Class", Springer-Verlag, 2009</li> <li>• Savitch, Walter, "Java. An Introduction to Problem Solving and Programming", Pearson, 6. Auflage, 2012</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 102801 Vorlesung Programmierung und Softwareentwicklung</li> <li>• 102802 Übung Programmierung und Softwareentwicklung</li> </ul>		

16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	63 h
	Selbststudiums- /	187 h
	Nachbearbeitungszeit:	
	Prüfungsvorbereitung:	20 h
	Summe:	270 h
<hr/>		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"><li>• 10281 Programmierung und Software-Entwicklung (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0,</li><li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich, Vorleistung: Übungsschein. Voraussetzungen werden zu Beginn vom Dozenten festgesetzt. Dazu gehören eine bestimmte Anzahl von Vorträgen in den Übungen und ein bestimmter Teil der Übungspunkte.</li></ul>	
<hr/>		
18. Grundlage für ... :	12060 Datenstrukturen und Algorithmen	
<hr/>		
19. Medienform:	<ul style="list-style-type: none"><li>• Folien über Beamer</li><li>• Tafelanschrieb</li></ul>	
<hr/>		
20. Angeboten von:	Software-Engineering	
<hr/>		

## Modul: 16520 Software-Qualität

2. Modulkürzel:	051520105	5. Moduldauer:	1 Semester
3. Leistungspunkte:	3.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	2.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Stefan Wagner		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Stefan Wagner</li> <li>• Ivan Bogicevic</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009, 1. Semester → Basismodule  B.Sc. Softwaretechnik, PO 2012, 1. Semester → Basismodule		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• gleichzeitiger Besuch der Programmierung und Softwareentwicklung</li> </ul>		
12. Lernziele:	Die Teilnehmer kennen und verstehen den Begriff der Software-Qualität. Insbesondere erfahren sie die Schwierigkeiten bei der Evolution großer Systeme. Sie kennen Techniken, deren Anwendung zu einer guten Software-Qualität beiträgt, und können sie anwenden.		
13. Inhalt:	In der Vorlesung wird der Begriff der Software-Qualität vermittelt und am Beispiel anschaulich gemacht. In der Übung wird ein großes Softwaresystem bearbeitet.		
14. Literatur:	<ul style="list-style-type: none"> <li>• Ludwig J., Lichter, H., Software Engineering - Grundlagen, Menschen, Prozesse, Techniken, 2007</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 165201 Vorlesung Software-Qualität</li> <li>• 165202 Übung Software-Qualität</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	21 h	
	Selbststudiums- /	69 h	
	Nachbearbeitungszeit:		
	Summe:	90 h	
17. Prüfungsnummer/n und -name:	16521 Software-Qualität (USL), schriftliche Prüfung, Gewichtung: 1.0, Übungsschein; Scheinkriterien werden zu Beginn der Veranstaltung angekündigt.		
18. Grundlage für ... :	17210	Einführung in die Softwaretechnik	
19. Medienform:			
20. Angeboten von:	Software-Engineering		

## Modul: 10940 Theoretische Grundlagen der Informatik

2. Modulkürzel:	050420005	5. Moduldauer:	2 Semester
3. Leistungspunkte:	12.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	8.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Apl. Prof. Ulrich Hertrampf		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Volker Diekert</li> <li>• Ulrich Hertrampf</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009, 1. Semester → Basismodule  B.Sc. Softwaretechnik, PO 2012, 1. Semester → Basismodule		
11. Empfohlene Voraussetzungen:	Keine		
12. Lernziele:	Logik und Diskrete Strukturen: <ul style="list-style-type: none"> <li>• Die Studierenden haben die grundsätzlichen Kenntnisse in Logik und Diskreter Mathematik erworben, wie sie in den weiteren Grundvorlesungen der Informatik in verschiedenen Bereichen benötigt werden.</li> </ul> Automaten und Formale Sprachen: <ul style="list-style-type: none"> <li>• Die Studierenden beherrschen wichtige theoretische Grundlagen der Informatik, insbesondere die Theorie und Algorithmik endlicher Automaten. Hierzu gehört das Kennenlernen, Einordnung und Trennung der Chomskyschen Sprachklassen.</li> </ul>		
13. Inhalt:	Logik und Diskrete Strukturen: <ul style="list-style-type: none"> <li>• Einführung in die Aussagenlogik: Semantik (Wahrheitswerte), Syntax (Axiome und Schlussregeln), Normalformen; Hornformeln; Endlichkeitssatz; aussagenlogische Resolution;</li> <li>• Einführung in die Prädikatenlogik 1. Stufe: Semantik und Syntax, Normalformen, Unifikatoren, Herbrand-Theorie, prädikatenlogische Resolution;</li> <li>• Elementare Zahlentheorie: Rechnen mit Restklassen, endliche Körper, Euklidischer Algorithmus, Chinesischer Restsatz, Primzahltests, RSA-Verfahren; Wachstumsabschätzungen; Grundbegriffe der Wahrscheinlichkeitsrechnung; Kombinatorik; Graphen.</li> </ul> Automaten und Formale Sprachen: <ul style="list-style-type: none"> <li>• Deterministische- bzw. nichtdeterministische endliche Automaten, reguläre Ausdrücke, Minimierung endlicher Automaten, Iterationslemmata für reguläre und kontextfreie Sprachen, Normalformen, Kellerautomaten, Lösen des Wortproblems kontextfreier Sprachen mit dem CYK-Algorithmus, linear beschränkte Automaten, kontextsensitive Grammatiken, Typ 0-Grammatiken und Turingmaschinen.</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• John Hopcroft, Jeffrey Ullman, Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie, 1988</li> <li>• Uwe Schöning, Theoretische Informatik - kurzgefasst, 1999</li> </ul>		

15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"><li>• 109401 Vorlesung Logik und Diskrete Strukturen</li><li>• 109402 Übung Logik und Diskrete Strukturen</li><li>• 109403 Vorlesung Automaten und Formale Sprachen</li><li>• 109404 Übung Automaten und Formale Sprachen</li></ul>								
16. Abschätzung Arbeitsaufwand:	<table><tr><td>Präsenzzeit:</td><td>84 h</td></tr><tr><td>Selbststudiums- /</td><td>276 h</td></tr><tr><td>Nachbearbeitungszeit:</td><td></td></tr><tr><td>Summe:</td><td>360 h</td></tr></table>	Präsenzzeit:	84 h	Selbststudiums- /	276 h	Nachbearbeitungszeit:		Summe:	360 h
Präsenzzeit:	84 h								
Selbststudiums- /	276 h								
Nachbearbeitungszeit:									
Summe:	360 h								
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"><li>• 10941 Theoretische Grundlagen der Informatik (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein</li><li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich, 30 Min.</li></ul>								
18. Grundlage für ... :									
19. Medienform:									
20. Angeboten von:	Institut für Formale Methoden der Informatik								

---

## 200 Kernmodule

---

Zugeordnete Module:	11890	Algorithmen und Berechenbarkeit
	14360	Einführung in die Technische Informatik
	14370	Fachstudie Softwaretechnik
	14390	Programmentwicklung
	14480	Sichere und zuverlässige Softwaresysteme
	16500	Software Engineering
	16510	Software-Praktikum
	36100	Programmierparadigmen
	41940	Studienprojekt-Th
	42790	Seminar SWT

---

## Modul: 11890 Algorithmen und Berechenbarkeit

2. Modulkürzel:	050420020	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Stefan Funke		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Stefan Funke</li> <li>• Volker Diekert</li> <li>• Ulrich Hertrampf</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009, 3. Semester → Kernmodule  B.Sc. Softwaretechnik, PO 2012, 3. Semester → Kernmodule		
11. Empfohlene Voraussetzungen:	Vorlesungen aus dem 1. und 2. Semester		
12. Lernziele:	Die Studierenden kennen die Klassifizierung von Algorithmen in effizient berechenbar, NP-vollständig, PSPACE-Algorithmen und prinzipielle Unberechenbarkeit. Sie haben wichtige Entwurfsstrategien und Analysemethoden kennengelernt.		
13. Inhalt:	Berechenbarkeit vs. Unberechenbarkeit, Church'sche These, NP-Vollständigkeit, PSPACE-vollständige Algorithmen (QBF). Entwurfsstrategien: Teile und Herrsche, gierig (greedy), Dynamisches Programmieren, Randomisierte Algorithmen		
14. Literatur:	<ul style="list-style-type: none"> <li>• John Hopcroft, Jeffrey Ullman: Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie, 1988</li> <li>• Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein: Introduction to Algorithms (Second Edition), 2001</li> <li>• Volker Diekert: Entwurf und Analyse effizienter Algorithmen (Vorlesungsskript), 2006</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 118901 Vorlesung Algorithmen und Berechenbarkeit</li> <li>• 118902 Übung Algorithmen und Berechenbarkeit</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- / Nacharbeitszeit:	138 h	
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 11891 Algorithmen und Berechenbarkeit (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein</li> <li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich, 30 Min.</li> </ul>		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Formale Methoden der Informatik		

## Modul: 14360 Einführung in die Technische Informatik

2. Modulkürzel:	051400105	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Otto Eggenberger		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Otto Eggenberger</li> <li>• Sven Simon</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009, 3. Semester → Kernmodule  B.Sc. Softwaretechnik, PO 2012, 3. Semester → Kernmodule		
11. Empfohlene Voraussetzungen:	Keine		
12. Lernziele:	Der/die Studierende kennt die grundlegende Funktionsweise eines Computers, versteht die elektrotechnischen Grundlagen und Technologien und kann einfache digitale Schaltungen analysieren, entwerfen und optimieren.		
13. Inhalt:	Grundlegende Funktionsweise eines Computers  <ul style="list-style-type: none"> <li>• Informationsdarstellung</li> <li>• Zahlendarstellung und Codes</li> <li>• Digitale Grundbausteine</li> <li>• Logische Funktionen, Speicherelemente</li> <li>• Befehlsausführung, Programmablauf</li> </ul> Elektrotechnische Grundlagen  <ul style="list-style-type: none"> <li>• Physikalische Grundbegriffe</li> <li>• Elektrische Spannung, elektrischer Strom</li> <li>• Elektrische Netzwerke</li> <li>• Halbleiterbauelemente</li> <li>• Digitale Grundschaltungen</li> </ul> Digitale Schaltungen  <ul style="list-style-type: none"> <li>• Schaltnetzwerke</li> <li>• Boolesche Algebra und Schaltalgebra</li> <li>• Darstellung und Minimierung von Schaltfunktionen</li> <li>• Rückkopplung, Zustandsbegriff</li> <li>• Automaten und sequentielle Netzwerke</li> <li>• Digitale Standardschaltungen</li> <li>• Entwurfsmethodik</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Dirk W. Hoffman: Grundlagen der technischen Informatik, Hanser, 2007</li> <li>• Bernd Becker, Rolf Drechsler, Paul Molitor: Technische Informatik, Pearson Studium, 2005</li> <li>• Jörg Keller, Wolfgang J. Paul: Hardware Design, Teubner, 3. Aufl. 2005</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 143601 Vorlesung Einführung in die Technische Informatik</li> <li>• 143602 Gruppenübungen Einführung in die Technische Informatik</li> </ul>		

16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h
	Selbststudiums- /	138 h
	Nachbearbeitungszeit:	
	Summe:	180 h
<hr/>		
17. Prüfungsnummer/n und -name:	14361	Einführung in die Technische Informatik (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0
<hr/>		
18. Grundlage für ... :	36530	Rechnerorganisation 1
<hr/>		
19. Medienform:		
<hr/>		
20. Angeboten von:	Institut für Rechnergestützte Ingenieursysteme	
<hr/>		

## Modul: 14370 Fachstudie Softwaretechnik

2. Modulkürzel:	051520185	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes Semester
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:		Univ.-Prof. Stefan Wagner	
9. Dozenten:		Dozenten der Informatik	
10. Zuordnung zum Curriculum in diesem Studiengang:		B.Sc. Softwaretechnik, PO 2009, 6. Semester → Kernmodule B.Sc. Softwaretechnik, PO 2012, 6. Semester → Kernmodule	
11. Empfohlene Voraussetzungen:		<ul style="list-style-type: none"> <li>• Bestandene Prüfung "Software Engineering"</li> </ul>	
12. Lernziele:		Die Teilnehmer sind in der Lage, eine konkrete praktische Frage der Softwaretechnik, beispielsweise über die anzuwendende Methode oder das geeignete Werkzeug, zu analysieren und zu entscheiden und ihre Entscheidung angemessen zu präsentieren. Die Arbeit erfolgt in Dreiergruppen.	
13. Inhalt:		Eine Gruppe analysiert eine (im Allgemeinen aus der Praxis kommende) Frage auf der Basis der Literatur und eigener Untersuchungen, auch Befragungen, und präsentiert ihre Empfehlung mündlich und in Form eines Berichts.	
14. Literatur:			
15. Lehrveranstaltungen und -formen:		<ul style="list-style-type: none"> <li>• 143701 Praktikum Fachstudie Softwaretechnik</li> <li>• 143702 Teamarbeit an den beteiligten Instituten mit örtlicher fachlicher Betreuung</li> </ul>	
16. Abschätzung Arbeitsaufwand:		Präsenzzeit: 42 h Selbststudiums- / Nachbearbeitungszeit: 138 h Summe: 180 h	
17. Prüfungsnummer/n und -name:		14371 Fachstudie Softwaretechnik (USL), schriftlich und mündlich, Gewichtung: 1.0	
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:		Software-Engineering	

## Modul: 14390 Programmentwicklung

2. Modulkürzel:	051520120	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:		Univ.-Prof. Stefan Wagner	
9. Dozenten:		<ul style="list-style-type: none"> <li>• Stefan Wagner</li> <li>• Jan-Peter Ostberg</li> </ul>	
10. Zuordnung zum Curriculum in diesem Studiengang:		B.Sc. Softwaretechnik, PO 2009, 3. Semester → Kernmodule  B.Sc. Softwaretechnik, PO 2012, 3. Semester → Kernmodule	
11. Empfohlene Voraussetzungen:		<ul style="list-style-type: none"> <li>• Programmierung und Softwareentwicklung</li> <li>• Einführung in die Softwaretechnik</li> </ul>	
12. Lernziele:		Die Teilnehmer kennen die Prinzipien der objektorientierten Programmierung und sind in der Lage, Programme in UML zu beschreiben und in Java zu implementieren.	
13. Inhalt:		<ul style="list-style-type: none"> <li>• Grundlagen der objektorientierten Programmierung</li> <li>• Spezifikation und Entwurf objektorientierter Programme mit UML</li> <li>• Vertiefte Programmierung in Java</li> </ul>	
14. Literatur:		<ul style="list-style-type: none"> <li>• Rumbaugh, Jacobson, Booch, The unified modeling language reference manual, 2nd ed., 2004</li> <li>• Rupp, Queins, Zengler, UML 2 glasklar: Praxiswissen für die UML-Modellierung, 3. Aufl. 2007</li> <li>• Ullenboom: Java ist auch eine Insel. Galileo Computing, 8. Aufl. 2009</li> </ul>	
15. Lehrveranstaltungen und -formen:		<ul style="list-style-type: none"> <li>• 143901 Vorlesung Programmentwicklung</li> <li>• 143902 Übung Programmentwicklung</li> </ul>	
16. Abschätzung Arbeitsaufwand:		Präsenzzeit: 42 h Selbststudiums- / Nachbearbeitungszeit: 138 h Summe: 180 h	
17. Prüfungsnummer/n und -name:		14391 Programmentwicklung (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0, Klausur 60 min, keine Vorleistungen.	
18. Grundlage für ... :			
19. Medienform:		<ul style="list-style-type: none"> <li>• Folien am Beamer unterstützt durch Tafel und Overhead</li> <li>• Dokumente, Links und Diskussionsforen in ILIAS</li> </ul>	
20. Angeboten von:		Software-Engineering	

## Modul: 36100 Programmierparadigmen

2. Modulkürzel:	051510010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Erhard Plödereder		
9. Dozenten:	Erhard Plödereder		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 4. Semester → Kernmodule		
11. Empfohlene Voraussetzungen:	Programmiererfahrung in mindestens einer Programmiersprache, vornehmlich Java, so wie z. B. im Modul "Programmierung und Software Entwicklung" (10280) erworben.		
12. Lernziele:	Die Studierenden haben grundlegende Konzepte von Programmiersprachen verstanden, die dem Erlernen weiterer Sprachen und dem vertieften Verständnis ihnen bekannter Sprachen dienlich sind. Sie haben deren Anwendung in mindestens einer weiteren Programmiersprache ihrer Ausbildung verstanden. Sie können ihre Kenntnisse in einfachen Programmen anwenden. Sie können weitere Programmiersprachen in ihrer akademischen und beruflichen Karriere schneller und präziser erlernen.		
13. Inhalt:	<p>Überblick typischer Konzepte in Programmiersprachen und ihrer Auswirkungen auf die Sprache und deren Anwendung:</p> <ul style="list-style-type: none"> <li>• Grundsätzliche Ausführungsmodelle</li> <li>• Speichermodelle und deren Konsequenzen</li> <li>• Datentypen und Typsysteme</li> <li>• unterschiedliche Bindungskonzepte und ihre Auswirkungen</li> <li>• objekt-orientierte Sprachkonzepte</li> <li>• Abstraktion und Kompositionsmechanismen</li> <li>• funktionale Sprachen</li> <li>• Eventuell werden auch Elemente der parallelen Programmierung und der Logik-Programmierung mit einbezogen.</li> </ul> <p>Die Vorlesung ist kein Streifzug durch diverse Programmiersprachen, sondern die Vorstellung zugrundeliegender Prinzipien, und ihrer Begründung aus der Sicht des Software Engineering, insbesondere der Zuverlässigkeit der Anwendung, und, wo nötig, der Implementierungsmodelle.</p>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Sebesta, Robert: Concepts of Programming Languages, Pearson Verlag, 11. ed., 2016 (Hörerschein verfügbar)</li> <li>• weitere Literatur wird zu Beginn der Lehrveranstaltung und auf den Webseiten bekanntgegeben.</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 361001 Vorlesung Programmierparadigmen</li> <li>• 361002 Übung Programmierparadigmen</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	

17. Prüfungsnummer/n und -name:
- 36101 Programmierparadigmen (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0
  - V Vorleistung (USL-V), schriftlich, eventuell mündlich
- 

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von: Institut für Softwaretechnologie

---

## Modul: 42790 Seminar SWT

2. Modulkürzel:	050420095	5. Moduldauer:	1 Semester
3. Leistungspunkte:	3.0 LP	6. Turnus:	jedes Semester
4. SWS:	2.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Stefan Wagner		
9. Dozenten:	Dozenten der Informatik		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Kernmodule		
11. Empfohlene Voraussetzungen:	Basismodule der Informatik, darüber hinaus variabel: Je nach dem gewählten Seminarthema können Vorkenntnisse aus weiteren Vorlesungen benötigt werden.		
12. Lernziele:	Die Studierenden können sich mit wissenschaftlicher Originalliteratur auseinandersetzen, deren Kernaussagen rezipieren und sich ein spezielles Thema überwiegend im Selbststudium erarbeiten. Sie sind fähig relevante Daten zu sammeln und zu interpretieren und ihre Erkenntnisse einem Fach- und Laienpublikum verständlich zu präsentieren und auf Fragen aus dem Publikum angemessen und sachgerecht zu reagieren. Sie haben gelernt, sich mit einem wissenschaftlichen Thema über einen längeren Zeitraum hinweg auseinander zu setzen und eigenständig aktuelle Hintergrundinformation zu beschaffen. Sie haben generische Kompetenzen erworben, etwa aktiv an einer wissenschaftlichen Diskussion zu einem vorher bekannten Thema teilzunehmen und durch Fragen an den Vortragenden ihr Verständnis zu erweitern. Sie können eine Diskussion leiten und moderieren und sind befähigt, ihre Ergebnisse den Seminarteilnehmern vorzustellen und mit Hilfe moderner Präsentationstechniken zu visualisieren. Sie sind in der Lage, das von ihnen erarbeitete Thema auch schriftlich darzustellen.		
13. Inhalt:	Variabel: Es werden Seminare zu diversen, häufig aktuellen Themen angeboten.  Welche Seminare zugelassen sind, entscheidet die Studienkommission. Zugelassene Seminare werden typischer Weise durch Aushang bekannt gegeben. Die Seminare sind in Größe und Inhalt so gestaltet, dass die generischen Kompetenzen (Schlüsselqualifikationen) der Studierenden entwickelt werden.		
14. Literatur:	Die begleitende Literatur wird in der Veranstaltung und im Web bekannt gegeben.		
15. Lehrveranstaltungen und -formen:	427901 Seminar		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	21 h	
	Selbststudiums- /	69 h	
	Nachbearbeitungszeit:		
	Summe:	90 h	
17. Prüfungsnummer/n und -name:	42791 Seminar SWT (BSL), Sonstiges, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			

20. Angeboten von:

Institut für Softwaretechnologie

---

## Modul: 14480 Sichere und zuverlässige Softwaresysteme

2. Modulkürzel:	051520115	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	André Hoorn		
9. Dozenten:	André Hoorn		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009, 5. Semester → Kernmodule B.Sc. Softwaretechnik, PO 2012, 5. Semester → Kernmodule		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>Teilnahme an der Lehrveranstaltung "Einführung in die Softwaretechnik"</li> </ul>		
12. Lernziele:	Die Teilnehmer beherrschen Techniken, deren Anwendung die Zuverlässigkeit der Software verbessert oder garantiert. Dabei stehen formale Ansätze im Vordergrund.		
13. Inhalt:	Notationen und Verfahren zur formalen Beschreibung und Prüfung der Software Notationen und Verfahren zum Sicherheits- Performanz- und Zuverlässigkeitsnachweis Verfahren zur Erstellung von sicheren und zuverlässigen Systemen		
14. Literatur:	A. Alessandro Birolini, Reliability Engineering, Springer Verlag Berlin Heidelberg 2010. B. Nancy G. Leveson, Safeware: System Safety and Computers. Addison-Wesley, 1995. C. Nancy G. Leveson, Engineering a Safer World: Systems Thinking Applied to Safety. MIT Press, 2011.		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>144801 Vorlesung Sichere und zuverlässige Softwaresysteme</li> <li>144802 Übung Sichere und zuverlässige Softwaresysteme</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	14481 Sichere und zuverlässige Softwaresysteme (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Zuverlässige Softwaresysteme		

## Modul: 16500 Software Engineering

2. Modulkürzel:	051520110	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	André Hoorn		
9. Dozenten:	André Hoorn		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009, 4. Semester → Kernmodule B.Sc. Softwaretechnik, PO 2012, 4. Semester → Kernmodule		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Einführung in die Softwaretechnik</li> <li>• Programmentwicklung</li> </ul>		
12. Lernziele:	Die Teilnehmer haben tiefe und umfassende Kenntnisse auf dem Gebiet des Softwareprojekt-Managements und in den Techniken der Software-Bearbeitung.		
13. Inhalt:	Ergänzend zur "Einführung in die Softwaretechnik" und daran anknüpfend behandelt diese Lehrveranstaltung folgende Themen: <ul style="list-style-type: none"> <li>• Softwarequalitätssicherung</li> <li>• Organisationsaspekte der Software-Bearbeitung</li> <li>• Software-Prozesse, Prozess-Bewertung und -Verbesserung</li> <li>• Software-Wartung</li> <li>• Weitere ausgewählte Kapitel des Software Engineerings</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Ludewig J., Lichter, H., Software Engineering - Grundlagen, Menschen, Prozesse, Techniken, 2. Aufl. 2010</li> <li>• Liggesmeyer P., Software-Qualität. Testen, Analysieren und Verifizieren von Software. Spektrum Akademischer Verlag, 2002.</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 165001 Vorlesung Software Engineering</li> <li>• 165002 Übung Software Engineering</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	16501 Software Engineering (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0,		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Softwaretechnologie		

## Modul: 16510 Software-Praktikum

2. Modulkürzel:	051520180	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Stefan Wagner		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Stefan Wagner</li> <li>• Jan-Peter Ostberg</li> <li>• Ivan Bogicevic</li> <li>• Jasmin Ramadani</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009, 3. Semester → Kernmodule  B.Sc. Softwaretechnik, PO 2012, 3. Semester → Kernmodule		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Einführung in die Softwaretechnik</li> <li>• Gleichzeitiger Besuch der Programmentwicklung</li> </ul>		
12. Lernziele:	Die Teilnehmer können eine Software-Entwicklung von der Spezifikation bis zur Auslieferung durchführen.		
13. Inhalt:	Die Teilnehmer bearbeiten in Dreiergruppen eine zentral gestellte Aufgabe. Sie erheben dazu die notwendigen Informationen, erstellen die notwendigen Dokumente und implementieren und prüfen ein Programm, das die Aufgabe löst.		
14. Literatur:	<ul style="list-style-type: none"> <li>• Ludwig J., Lichter, H., Software Engineering - Grundlagen, Menschen, Prozesse, Techniken. dpunkt-Verlag, Heidelberg, 2. Aufl. 2010</li> </ul>		
15. Lehrveranstaltungen und -formen:	165101 Praktikum Software-Praktikum		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	16511 Software-Praktikum (USL), Sonstiges, Gewichtung: 1.0		
18. Grundlage für ... :	<ul style="list-style-type: none"> <li>• 16610 Studienprojekt-Pr</li> <li>• 16780 Studienprojekt-Th</li> </ul>		
19. Medienform:	Die meisten Dokumente erarbeiten die Studierenden selbst und stellen sie auch vor. Zusatzinformationen und Diskussionsforen werden in ILIAS zur Verfügung gestellt.		
20. Angeboten von:	Software-Engineering		

## Modul: 41940 Studienprojekt-Th

2. Modulkürzel:	051520192	5. Moduldauer:	2 Semester
3. Leistungspunkte:	9.0 LP	6. Turnus:	jedes Semester
4. SWS:	6.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Stefan Wagner		
9. Dozenten:	Dozenten der Informatik		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 4. Semester → Kernmodule		
11. Empfohlene Voraussetzungen:	Die Module Einführung in die Softwaretechnik, Programmentwicklung und Software-Praktikum müssen vor Beginn des Studienprojekts absolviert sein. Das Studienprojekt-Th bildet mit dem Studienprojekt-Pr eine Einheit; beide können nur zusammen begonnen werden. Die Vorleistungen (Scheine) aus dem Studienprojekt-Pr sind für die abschließende Prüfung des Studienprojekt-Th Voraussetzung.		
12. Lernziele:	Vorlesung und Seminar dienen dazu, theoretische Grundlagen zum Studienprojekt-Pr zu vermitteln und die Arbeit im Projekt zu reflektieren.		
13. Inhalt:	Die konkreten Themen und Inhalte hängen vom jeweiligen Projekt ab.		
14. Literatur:	Deiningner, Lichter, Ludewig, Schneider , Studien-Arbeiten, 5. Aufl. 2005		
15. Lehrveranstaltungen und -formen:			
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	63 h	
	Selbststudiums- /	207 h	
	Nachbearbeitungszeit:		
	Summe:	270 h	
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 41941 Studienprojekt-Th (PL), mündliche Prüfung, 25 Min., Gewichtung: 1.0</li> <li>• 41942 Studienprojekt-Th - benotete Studienleistung (BSL), schriftlich und mündlich, Gewichtung: 1.0</li> </ul>		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Software-Engineering		

## 300 Ergänzungsmodule

---

Zugeordnete Module:	310	Katalog SWT
	320	Katalog ISG
	330	Katalog ISW
	340	Wahlmodule aus Master SWT

---

## 310 Katalog SWT

---

Zugeordnete Module:    10210 Mensch-Computer-Interaktion  
                              10220 Modellierung  
                              36530 Rechnerorganisation 1  
                              40090 Systemkonzepte und -programmierung  
                              41590 Einführung in die Numerik und Stochastik für Softwaretechniker

---

## Modul: 41590 Einführung in die Numerik und Stochastik für Softwaretechniker

2. Modulkürzel:	051240006	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Dirk Pflüger		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Dirk Pflüger</li> <li>• Stefan Zimmer</li> <li>• Thomas Ertl</li> <li>• Miriam Mehl</li> <li>• Daniel Weiskopf</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 5. Semester          → Ergänzungsmodule --&gt;Katalog ISG          →</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester          → Ergänzungsmodule --&gt;Katalog ISW          →</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester          → Ergänzungsmodule --&gt;Katalog SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Katalog ISG          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Katalog ISW          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Katalog SWT          →</p>		
11. Empfohlene Voraussetzungen:	Mathematik für Informatiker und Softwaretechniker (Modulnummer 10190)		
12. Lernziele:	Beherrschung grundlegender Begriffe und Methoden der Numerik und Stochastik, Kenntnis der Anwendungsbereiche und Gültigkeitsgrenzen der erlernten Methoden, insbesondere Kenntnis der Auswirkungen von Näherungen, Beherrschung der Modellierung einfacher Probleme mit stochastischen Methoden.		
13. Inhalt:	<p>Methoden der angewandten Mathematik, insbesondere der Numerik und Stochastik, sind für viele Bereiche der Informatik wie Simulation, Grafik oder Bildverarbeitung von zentraler Bedeutung. In Ergänzung der Mathematik-Grundausbildung vermittelt diese Vorlesung folgende Grundkenntnisse:</p> <ul style="list-style-type: none"> <li>• numerische Algorithmik</li> <li>• Gleitpunktzahlen und Gleitpunktarithmetik</li> <li>• Interpolation &amp; Approximation</li> <li>• Integration</li> <li>• lineare Gleichungssysteme</li> <li>• Iterative Lösung linearer und nichtlinearer Gleichungen</li> <li>• gewöhnliche Differentialgleichungen</li> <li>• Stochastik</li> </ul>		

- Zufall und Unsicherheit
  - diskrete und kontinuierliche Wahrscheinlichkeitsräume
  - Asymptotik
- 

14. Literatur:

- Dahmen, Reusken: Numerik für Ingenieure
  - Schwarz, Köckler: Numerische Mathematik
  - Huckle, Schneider: Numerik für Informatiker
  - Henze: Stochastik für Einsteiger
  - Schickinger, Steger: Diskrete Strukturen, Band 2
- 

15. Lehrveranstaltungen und -formen:

- 415901 Vorlesung Einführung in die Numerik und Stochastik für Softwaretechniker
  - 415902 Übung Einführung in die Numerik und Stochastik für Softwaretechniker
- 

16. Abschätzung Arbeitsaufwand:

Präsenzzeit:	42 h
Selbststudiums- /	138 h
Nachbearbeitungszeit:	
Summe:	180 h

---

17. Prüfungsnummer/n und -name:

41591 Einführung in die Numerik und Stochastik für Softwaretechniker (PL), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 1.0

---

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von:

Simulation großer Systeme

---

## Modul: 10210 Mensch-Computer-Interaktion

2. Modulkürzel:	051900001	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Albrecht Schmidt		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Albrecht Schmidt</li> <li>• Thomas Ertl</li> <li>• Daniel Weiskopf</li> <li>• Niels Henze</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog SWT →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog SWT →</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 10280 Programmierung und Software-Entwicklung</li> </ul>		
12. Lernziele:	<p>Studierende entwickeln ein Verständnis für Modelle, Methoden und Konzepte der Mensch-Computer-Interaktion. Sie lernen verschiedene Ansätze für den Entwurf, die Entwicklung und Bewertung von Benutzungsschnittstellen kennen und verstehen deren Vor- und Nachteile.</p>		
13. Inhalt:	<p>Die Vorlesung vermittelt Konzepte, Prinzipien, Modelle, Methoden und Techniken für die effektive Entwicklung von benutzerfreundlichen Mensch-Computer-Schnittstellen. Das Thema moderner Benutzungsschnittstellen wird dabei für klassische Computer aber auch für mobile Geräte, eingebettete Systeme, Automobile und intelligente Umgebungen betrachtet.</p> <p>Die folgenden Themen werden in der Vorlesung behandelt:</p> <ul style="list-style-type: none"> <li>• Einführung in die Grundlagen der Mensch-Computer Interaktion, historische Entwicklung</li> <li>• Entwurfsprinzipien und Modelle für moderne Benutzungsschnittstellen und interaktive Systeme</li> <li>• Informationsverarbeitung des Menschen, Wahrnehmung, Motorik, Eigenschaften und Fähigkeiten des Benutzers</li> <li>• Interaktionskonzepte und -stile, Metaphern, Normen, Regeln und Style Guides</li> </ul>		

- Ein- und Ausgabegeräte, Entwurfsraum für interaktive Systeme
  - Analyse-, Entwurfs- und Entwicklungsmethoden und -werkzeuge für Benutzungsschnittstellen
  - Prototypische Realisierung und Implementierung von interaktiven Systemen, Werkzeuge
  - Architekturen für interaktive Systeme, User Interface Toolkits und Komponenten
  - Akzeptanz, Evaluationsmethoden und Qualitätssicherung
- 

14. Literatur:
- Bernhard Preim, Raimund Dachsel. Interaktive Systeme 1: Grundlagen, Graphical User Interfaces, Informationsvisualisierung. Springer, Berlin; 2. Auflage. 2010
  - Alan Dix, Janet Finley, Gregory Abowd, Russell Beale, Human-Computer Interaction, 2004
  - Ben Shneiderman, Catherine Plaisant, Designing the User Interfaces, 2005
- 

15. Lehrveranstaltungen und -formen:
- 102101 Vorlesung Mensch-Computer-Interaktion
  - 102102 Übung Mensch-Computer-Interaktion
- 

16. Abschätzung Arbeitsaufwand:
- |                       |       |
|-----------------------|-------|
| Präsenzzeit:          | 42 h  |
| Selbststudiums- /     | 138 h |
| Nachbearbeitungszeit: |       |
| Summe:                | 180 h |
- 

17. Prüfungsnummer/n und -name:
- 10211 Mensch-Computer-Interaktion (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein
  - V Vorleistung (USL-V), schriftlich, eventuell mündlich
- 

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von: Institut für Visualisierung und Interaktive Systeme

---

## Modul: 10220 Modellierung

2. Modulkürzel:	052010001	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Bernhard Mitschang</li> <li>• Frank Leymann</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog SWT →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog SWT →</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• 051520005 Programmierung und Software-Entwicklung</li> <li>• 051510005 Datenstrukturen und Algorithmen</li> <li>• 051200005 Systemkonzepte und -programmierung</li> </ul>		
12. Lernziele:	<p>Am Ende des Moduls sind die Studierenden in der Lage, wesentliche Artefakte eines IT Systems zu modellieren. Der Zusammenhang und das Zusammenspiel solcher Artefakte ist verstanden. Die Rolle von Metamodellen und deren Erstellung ist klar.</p>		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Entity-Relationship Modell &amp; komplexe Objekte</li> <li>• Relationenmodell &amp; Relationenalgebra , Überblick SQL</li> <li>• Transformationen von ER nach Relationen, Normalisierung</li> <li>• XML, DTD, XML-Schema, Info-Set, Namensräume</li> <li>• Metamodelle &amp; Repository</li> <li>• RDF, RDF-S &amp; Ontologien</li> <li>• UML</li> <li>• Petri Netze, Workflownetze</li> <li>• BPMN</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• A. Silberschatz, H. F. Korth, S. Sudarshan, Database System Concepts, 2002</li> <li>• R. Eckstein, S. Eckstein, "XML und Datenmodellierung", dpunkt.verlag 2004</li> <li>• M. Hitz, G. Kappel, E. Kapsammer, W. Retschitzegger, UML @ Work - Objektorientierte Modellierung mit UML2, 2005</li> <li>• P. Hitzler, M. Krötzsch, S. Rudolph, Y. Sure, Semantic Web, 2008</li> <li>• T.J. Teorey, Database Modeling &amp; Design, 2nd Edition, 1994</li> </ul>		

- H.J. Habermann, F. Leymann, "Repository", Oldenbourg 1993
- W. Reisig, "Petri-Netze", Vieweg & Teubner 2010
- B. Silver, "BPMN Method & Style", Cody-Cassidy Press 2009

---

15. Lehrveranstaltungen und -formen:

- 102201 Vorlesung Modellierung
- 102202 Übung Modellierung

---

16. Abschätzung Arbeitsaufwand:

Präsenzzeit: 42 Stunden  
Nachbearbeitungszeit: 138 Stunden

---

17. Prüfungsnummer/n und -name:

- 10221 Modellierung (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein
- V Vorleistung (USL-V), schriftlich, eventuell mündlich

---

18. Grundlage für ... :

- 10030 Architektur von Anwendungssystemen
- 10080 Datenbanken und Informationssysteme

---

19. Medienform:

---

20. Angeboten von:

Institut für Architektur von Anwendungssystemen

---

## Modul: 36530 Rechnerorganisation 1

2. Modulkürzel:	051700006	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Hans-Joachim Wunderlich		
9. Dozenten:	Hans-Joachim Wunderlich		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog SWT →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog SWT →</p>		
11. Empfohlene Voraussetzungen:	Einführung in die Technische Informatik (14360)		
12. Lernziele:	<ul style="list-style-type: none"> <li>• Kenntnisse über die Grundlagen der Organisation von Rechnern und deren maschinennahe Programmierung</li> <li>• Grundzüge über die Beschreibung und den Entwurf von Hardwaresystemen</li> </ul>		
13. Inhalt:	<p>Aufbau, Entwurf und maschinennahe Programmierung von Mikroprozessoren werden erläutert. In den Übungen wird das Wissen durch schriftliche Aufgaben sowie Experimente mit Prozessorsimulatoren vertieft. Im einzelnen werden behandelt:</p> <ul style="list-style-type: none"> <li>• Grundstrukturen: Stack-, Akkumulator- und Register-basierende Rechner</li> <li>• Informationsdarstellung in Hardware: Daten- und Befehlsformate, Fehlererkennung und -korrektur, Befehlssätze und Unterstützung von Hochsprachen.</li> <li>• MIPS als RISC-Bespiel und seine Assemblerprogrammierung</li> <li>• Grundelemente und Entwurf auf Register-Transfer-Ebene</li> <li>• Grundzüge einer Hardware-Beschreibungssprache</li> <li>• Operationswerke: Multiplikation, Division, Gleitkommaeinheiten</li> <li>• Steuerwerksentwurf und Mikroprogrammierung</li> <li>• Befehlszyklus und Unterbrechungen</li> <li>• Pipelining und statisches Scheduling</li> <li>• Speicherorganisation: Cachestrukturen und virtueller Speicher, Seitenverwaltung, Segmentierung, TLB, MMU und DMA</li> <li>• Leistungsbewertung: Maßzahlen und CPI, Benchmarking und einfache Anwendung von Warteschlangen</li> </ul>		

14. Literatur:	Literatur, siehe Webseite zur Veranstaltung								
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"><li>• 365301 Vorlesung Rechnerorganisation 1</li><li>• 365302 Praktische und theoretische Übungen Rechnerorganisation 1</li></ul>								
16. Abschätzung Arbeitsaufwand:	<table><tr><td>Präsenzzeit:</td><td>42 h</td></tr><tr><td>Selbststudiums- /</td><td>138 h</td></tr><tr><td>Nachbearbeitungszeit:</td><td></td></tr><tr><td>Summe:</td><td>180 h</td></tr></table>	Präsenzzeit:	42 h	Selbststudiums- /	138 h	Nachbearbeitungszeit:		Summe:	180 h
Präsenzzeit:	42 h								
Selbststudiums- /	138 h								
Nachbearbeitungszeit:									
Summe:	180 h								
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"><li>• 36531 Rechnerorganisation 1 (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0</li><li>• V Vorleistung (USL-V), schriftliche Prüfung</li></ul>								
18. Grundlage für ... :									
19. Medienform:									
20. Angeboten von:	Rechnerarchitektur								

## Modul: 40090 Systemkonzepte und -programmierung

2. Modulkürzel:	051200005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Kurt Rothermel		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Kurt Rothermel</li> <li>• Frank Dürr</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog SWT →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog SWT →</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>* Modul 051520005 Programmierung und Software-Entwicklung</li> <li>* Modul 051510005 Datenstrukturen und Algorithmen</li> </ul>		
12. Lernziele:	<ul style="list-style-type: none"> <li>* Verstehen grundlegender Architekturen und Organisationsformen von Software-Systemen</li> <li>* Verstehen systemnaher Konzepte und Mechanismen</li> <li>* Kann existierende Systemplattformen und Betriebssysteme hinsichtlich ihrer Eigenschaften analysieren und anwenden.</li> <li>* Kann systemnahe Software entwerfen und implementieren.</li> <li>* Kann nebenläufige Programme entwickeln</li> <li>* Kann mit Experten anderer Fachgebiete die Anwendung von Systemfunktionen abstimmen.</li> </ul>		
13. Inhalt:	<p>Grundlegende Systemstrukturen - und organisationen</p> <ul style="list-style-type: none"> <li>• Multitaskingsystem</li> <li>• Multiprozessorsystem</li> <li>• Verteiltes System</li> </ul> <p>Modellierung und Analyse nebenläufiger Programme</p> <ul style="list-style-type: none"> <li>• Abstraktionen: Atomare Befehle, Prozesse, nebenläufiges Programm</li> <li>• Korrektheit- und Leitungskriterien</li> </ul> <p>Betriebssystemkonzepte</p> <ul style="list-style-type: none"> <li>• Organisation von Betriebssystemen</li> <li>• Prozesse und Threads</li> <li>• Eingabe/Ausgabe</li> <li>• Scheduling</li> </ul>		

Konzepte zur Synchronisation über gemeinsamen Speicher

- Synchronisationsprobleme und -lösungen
- Synchronisationswerkzeuge: Semaphor, Monitor

Konzepte zur Kommunikation und Synchronisation mittels Nachrichtentransfer

- Taxonomie: Kommunikation und Synchronisation
- Nachrichten als Kommunikationskonzept
- Höhere Kommunikationskonzepte

Basialgorithmen für Verteilte Systeme

- Erkennung globaler Eigenschaften
- Schnappschussproblem
- Konsistenter globaler Zustand
- Verteilte Terminierung

Praktische nebenläufige Programmierung in Java

- Threads und Synchronisation
- Socketschnittstelle
- RMI Programmierung

---

14. Literatur:	Literatur, siehe Webseite zur Veranstaltung
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"><li>• 400901 Vorlesung Systemkonzepte und -programmierung</li><li>• 400902 Übung Systemkonzepte und -programmierung</li></ul>
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"><li>• 40091 Systemkonzepte und -programmierung (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0</li><li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich</li></ul>
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Verteilte Systeme

---

## 320 Katalog ISG

---

Zugeordnete Module:	10030	Architektur von Anwendungssystemen
	10060	Computergraphik
	10100	Grundlagen der Graphischen Ingenieursysteme
	10110	Grundlagen der Künstlichen Intelligenz
	10170	Imaging Science
	10180	Information Retrieval und Text Mining
	10210	Mensch-Computer-Interaktion
	10220	Modellierung
	18560	Grundlagen der Eingebetteten Systeme
	36530	Rechnerorganisation 1
	39040	Rechnernetze
	40090	Systemkonzepte und -programmierung
	41590	Einführung in die Numerik und Stochastik für Softwaretechniker
	42410	Grundlagen des Wissenschaftlichen Rechnens
	56930	Grundlagen der Rechnerarchitektur

---

## Modul: 10030 Architektur von Anwendungssystemen

2. Modulkürzel:	052010002	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	Frank Leymann		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p>		
11. Empfohlene Voraussetzungen:	Vorlesungen des Grundstudiums.		
12. Lernziele:	<p>Die Vorlesung erläutert den Begriff der Architektur von Anwendungssystemen und die Rolle des Architekten solcher Systeme. Die wesentlichen Bestandteile von Anwendungsarchitektur wie etwa Datenbanksysteme, Anwendungsserver, Messaging Systeme, Workflowsysteme und TP-Monitore werden diskutiert. Die wesentlichen Mustern zur Erstellung von Anwendungssystemen sind verstanden.</p>		
13. Inhalt:	<p>Architekturelle Stile wie etwa N-stufige Aufbauten oder Service-Orientierung werden vorgestellt. Architekturmuster werden detailliert. Fundamentale Konzepte wie Transaktionen und Queuing werden eingeführt. Darauf aufbauend wird Direct TP vs Queues TP diskutiert. Grundlegende Qualitätseigenschaften wie Verfügbarkeit und Skalierbarkeit werden erläutert und Mechanismen zu deren Erzielen eingeführt. Die Rolle von Komponenten und Programmierung im Großen wird heraus gearbeitet und Modell-getriebene Architektur vorgestellt.</p>		
14. Literatur:	<ul style="list-style-type: none"> <li>• A. Silberschatz, H. F. Korth, S. Sudarshan, Database System Concepts, 2002</li> <li>• B. Neubauer, T. Ritter, F. Stoinnski, CORBA Komponenten, 2004</li> <li>• F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal, Pattern-orientierte Software Architektur - Ein Patternsystem, 1998</li> <li>• F. Leymann, D. Roller, Production Workflow, 2000</li> <li>• L. Hohmann, Beyond Software Architecture, 2003</li> <li>• M. Fowler, Patters of Enterprise Application Architecture, 2003</li> <li>• P. Bernstein, E. Newcomer, Principles of Transaction Processing, 1997</li> <li>• S. Conrad, W. Hasselbring, A. Koschel, R. Tritsch, Enterprise Application Integration, 2006</li> <li>• S. Weerawarana, F. Curbera, F. Leymann, T. Storey, D. Ferguson, Web Services Platform Architecture, 2005</li> </ul>		

	<ul style="list-style-type: none"><li>• W. Emmerich, Konstruktion von verteilten Objekten, 2003</li></ul>
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"><li>• 100301 Vorlesung Grundlagen der Architektur von Anwendungssystemen</li><li>• 100302 Übung Grundlagen der Architektur von Anwendungssystemen</li></ul>
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"><li>• 10031 Architektur von Anwendungssystemen (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0,</li><li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich</li></ul>
18. Grundlage für ... :	<ul style="list-style-type: none"><li>• 29480 Loose Coupling and Message Based Applications</li><li>• 29510 Service Computing</li><li>• 29530 Business Process Management</li><li>• 42520 Services and Service Composition</li></ul>
19. Medienform:	Vorlesungen mit begleitenden Übungen
20. Angeboten von:	Architektur von Anwendungssystemen

---

## Modul: 10060 Computergraphik

2. Modulkürzel:	051900002	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Thomas Ertl		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Thomas Ertl</li> <li>• Daniel Weiskopf</li> <li>• Martin Fuchs</li> <li>• Guido Reina</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 10210 Mensch-Computer-Interaktion</li> <li>• Modul 41590 Einführung in die Numerik und Stochastik</li> </ul>		
12. Lernziele:	Die Studierenden haben Wissen über die Grundlagen der Computergraphik sowie praktische Fähigkeiten in der Graphikprogrammierung erworben.		
13. Inhalt:	<p>Folgende Themen werden in der Vorlesung behandelt:</p> <ul style="list-style-type: none"> <li>• Überblick über den Prozess der Bildsynthese</li> <li>• Graphische Geräte, visuelle Wahrnehmung, Farbsysteme</li> <li>• Grundlegende Rastergraphik und Bildverarbeitung</li> <li>• Raytracing und Beleuchtungsmodelle</li> <li>• 2D und 3D Geometrietransformationen, 3D Projektion</li> <li>• Graphikprogrammierung in OpenGL 3</li> <li>• Texturen</li> <li>• Polygonale und hierarchische Modelle</li> <li>• Rasterisierung und Verdeckungsrechnung</li> <li>• Grundlagen der geometrischen Modellierung (Kurven, Flächen)</li> <li>• Räumliche Datenstrukturen</li> </ul> <p>Die Veranstaltung besteht aus Vorlesung mit Übungen. Die Übungen umfassen praktische Programmierübungen, theoretische Themen und Programmierprojekte.</p>		
14. Literatur:	<ul style="list-style-type: none"> <li>• J. Encarnacao, W. Strasser, R. Klein: Graphische Datenverarbeitung (Band1 und 2), 1997</li> <li>• J. Foley, A. van Dam, S. Feiner, J. Hughes: Computer Graphics: Principle and Practice, 1990</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 100601 Vorlesung Computergraphik</li> <li>• 100602 Übung Computergraphik</li> </ul>		

16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h
	Selbststudiums- /	138 h
	Nachbearbeitungszeit:	
	Summe:	180 h
<hr/>		
17. Prüfungsnummer/n und -name:	• 10061 Computergraphik (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein.	
	• V Vorleistung (USL-V), schriftlich, eventuell mündlich	
<hr/>		
18. Grundlage für ... :		
<hr/>		
19. Medienform:		
<hr/>		
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme	
<hr/>		

## Modul: 41590 Einführung in die Numerik und Stochastik für Softwaretechniker

2. Modulkürzel:	051240006	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Dirk Pflüger		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Dirk Pflüger</li> <li>• Stefan Zimmer</li> <li>• Thomas Ertl</li> <li>• Miriam Mehl</li> <li>• Daniel Weiskopf</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog SWT →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog SWT →</p>		
11. Empfohlene Voraussetzungen:	Mathematik für Informatiker und Softwaretechniker (Modulnummer 10190)		
12. Lernziele:	Beherrschung grundlegender Begriffe und Methoden der Numerik und Stochastik, Kenntnis der Anwendungsbereiche und Gültigkeitsgrenzen der erlernten Methoden, insbesondere Kenntnis der Auswirkungen von Näherungen, Beherrschung der Modellierung einfacher Probleme mit stochastischen Methoden.		
13. Inhalt:	<p>Methoden der angewandten Mathematik, insbesondere der Numerik und Stochastik, sind für viele Bereiche der Informatik wie Simulation, Grafik oder Bildverarbeitung von zentraler Bedeutung. In Ergänzung der Mathematik-Grundausbildung vermittelt diese Vorlesung folgende Grundkenntnisse:</p> <ul style="list-style-type: none"> <li>• numerische Algorithmik</li> <li>• Gleitpunktzahlen und Gleitpunktarithmetik</li> <li>• Interpolation &amp; Approximation</li> <li>• Integration</li> <li>• lineare Gleichungssysteme</li> <li>• Iterative Lösung linearer und nichtlinearer Gleichungen</li> <li>• gewöhnliche Differentialgleichungen</li> <li>• Stochastik</li> </ul>		

- Zufall und Unsicherheit
  - diskrete und kontinuierliche Wahrscheinlichkeitsräume
  - Asymptotik
- 

14. Literatur:

- Dahmen, Reusken: Numerik für Ingenieure
  - Schwarz, Köckler: Numerische Mathematik
  - Huckle, Schneider: Numerik für Informatiker
  - Henze: Stochastik für Einsteiger
  - Schickinger, Steger: Diskrete Strukturen, Band 2
- 

15. Lehrveranstaltungen und -formen:

- 415901 Vorlesung Einführung in die Numerik und Stochastik für Softwaretechniker
  - 415902 Übung Einführung in die Numerik und Stochastik für Softwaretechniker
- 

16. Abschätzung Arbeitsaufwand:

Präsenzzeit:	42 h
Selbststudiums- /	138 h
Nachbearbeitungszeit:	
Summe:	180 h

---

17. Prüfungsnummer/n und -name:

41591 Einführung in die Numerik und Stochastik für Softwaretechniker (PL), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 1.0

---

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von:

Simulation großer Systeme

---

## Modul: 18560 Grundlagen der Eingebetteten Systeme

2. Modulkürzel:	051711010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Martin Radetzki		
9. Dozenten:	Martin Radetzki		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISG → B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISW →		
11. Empfohlene Voraussetzungen:	Modul 051700005 Rechnerorganisation		
12. Lernziele:	Methodische und werkzeugunterstützte Konstruktion von Hardware und Software für auf spezifische Anwendungen hin optimierte eingebettete Systeme.		
13. Inhalt:	Einführung in Charakteristik und Anwendungsgebiete eingebetteter Systeme, hardwarenahe Softwareentwicklung, Software-Scheduling, eingebettete Algorithmen (digitale Signalverarbeitung, Kanal- und Quellencodierung am Beispiel Viterbi-Algorithmus und MPEG-Video-Codec), zustandsbasierte Modellierung (Statecharts), eingebettete Prozessoren (Microcontroller, digitale Signalprozessoren, ARM), Bussysteme und Speicher, Systemsynthese (Taskgraphen, Allokation, Bindung, Ablaufplanung/Scheduling)		
14. Literatur:	J. Teich, Digitale Hardware/Software-Systeme, 2. Auflage, 2007 P. Marwedel, Embedded System Design, 2006		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 185601 Vorlesung Grundlagen der Eingebetteten Systeme</li> <li>• 185602 Übung Grundlagen der Eingebetteten Systeme</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 18561 Grundlagen der Eingebetteten Systeme (PL), schriftliche Prüfung, 90 Min., Gewichtung: 75.0</li> <li>• V Vorleistung (USL-V), schriftliche Prüfung</li> </ul>		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Eingebettete Systeme (Embedded Systems Engineering)		

## Modul: 10100 Grundlagen der Graphischen Ingenieursysteme

2. Modulkürzel:	051400005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Dieter Roller		
9. Dozenten:	Dieter Roller		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p>		
11. Empfohlene Voraussetzungen:	Grundstudium		
12. Lernziele:	<ul style="list-style-type: none"> <li>• Kenntnis und Verständnis von Modellen bei der Produktentwicklung</li> <li>• Grundkenntnisse über die wichtigsten Modellarten, Algorithmen und Datenstrukturen und Techniken für den Datenaustausch</li> </ul>		
13. Inhalt:	<p>Inhalte:</p> <ul style="list-style-type: none"> <li>• Anforderungen an CAD-Systeme</li> <li>• zweidimensionale Modelle</li> <li>• dreidimensionale Modelle</li> <li>• interaktive Modellerstellung</li> <li>• Einführung in die Modifikationstechnik u. parametrische Modellierung</li> <li>• Methoden zur Modellmodifikation</li> <li>• Grundlagen der parametrischen Modellierung</li> <li>• Ansätze und Verfahren zur parametrischen Variantenerzeugung</li> <li>• Ausgewählte Anwendungsbeispiele</li> <li>• Überblick über weitergehende Modellieransätze</li> <li>• Datenverwaltung in CAD</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• D. Roller, CAD - Effiziente Anpassungs- und Variantenkonstruktion, Springer-Verlag</li> <li>• Literatur, siehe Webseite zur Veranstaltung</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 101001 Vorlesung Grundlagen der Graphischen Ingenieursysteme</li> <li>• 101002 Übung Grundlagen der Graphischen Ingenieursysteme</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	10101 Grundlagen der Graphischen Ingenieursysteme (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0		
18. Grundlage für ... :			

19. Medienform:

20. Angeboten von:

Institut für Rechnergestützte Ingenieursysteme

---

## Modul: 10110 Grundlagen der Künstlichen Intelligenz

2. Modulkürzel:	051900205	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Marc Toussaint		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Andrés Bruhn</li> <li>• Marc Toussaint</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 080300100 Mathematik für Informatiker und Softwaretechniker</li> </ul>		
12. Lernziele:	Der Student / die Studentin beherrscht die Grundlagen der Künstlichen Intelligenz, kann Probleme der KI selbständig einordnen und mit den erlernten Methoden und Algorithmen bearbeiten.		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Intelligenz</li> <li>• Agentenbegriff</li> <li>• Problemlösen durch Suchen, Suchverfahren</li> <li>• Probleme mit Rand- und Nebenbedingungen</li> <li>• Spiele</li> <li>• Aussagen- und Prädikatenlogik</li> <li>• Logikbasierte Agenten, Wissensrepräsentation</li> <li>• Inferenz</li> <li>• Planen</li> <li>• Unsicherheit, probabilistisches Schließen</li> <li>• Probabilistisches Schließen über die Zeit</li> <li>• Sprachverarbeitung</li> <li>• Entscheidungstheorie</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• S. Russell, P. Norvig, Künstliche Intelligenz, 2004</li> <li>• G. F. Luger, Künstliche Intelligenz, 2001</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 101101 Vorlesung Grundlagen der Künstlichen Intelligenz</li> <li>• 101102 Übung Grundlagen der Künstlichen Intelligenz</li> </ul>		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden</p> <p><b>Gesamt: 180 Stunden</b></p>		

17. Prüfungsnummer/n und -name:
- 10111 Grundlagen der Künstlichen Intelligenz (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben
  - V Vorleistung (USL-V), schriftlich, eventuell mündlich
- 

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von: Institut für Visualisierung und Interaktive Systeme

---

## Modul: 56930 Grundlagen der Rechnerarchitektur

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Hans-Joachim Wunderlich		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Michael Kochte</li> <li>• Hans-Joachim Wunderlich</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 41930 Rechnerorganisation</li> </ul>		
12. Lernziele:	<ul style="list-style-type: none"> <li>• Tiefes Verständnis von Entwurfskonzepten, die in modernen Prozessoren und Rechensystemen Verwendung finden</li> <li>• Kenntnis von Entwurfsherausforderungen.</li> <li>• Verständnis von aktuellen und zukünftigen Entwicklungstrends</li> </ul>		
13. Inhalt:	<p>Grundlegende und fortgeschrittene Themen der Rechnerarchitektur, inklusive:</p> <ul style="list-style-type: none"> <li>• Technologiegrundlagen: Entwurfsverfahren, Herstellungsmethoden, Ausbeute, Test und Zuverlässigkeit, Kosten und Qualität, Skalierung.</li> <li>• Rechen- und Verlustleistung: Analyse und Optimierung</li> <li>• Rechnerarithmetik: Effiziente Hardwarestrukturen für grundlegende Arithmetik, Implementierung von Logarithmen, Exponentialfunktion und trigonometrischen Funktionen, arithmetische Pipelines, praktische Implementierungen von Gleitkommaarithmetik (Cell SPE, SPARC).</li> <li>• Instruktionsparallelismus (ILP): Superskalarität, statisches und dynamisches Scheduling, out-of-order execution, VLIW Prozessoren, Multithreading.</li> <li>• Datenparallelismus (DLP): Vektorprozessoren, SIMD, Grafikprozessoreinheiten (GPGPU)</li> <li>• Threadparallelismus (TLP): Mehrprozessorsysteme, Speicherkohärenz und Synchronisierung.</li> <li>• Speicher- und Cache-Architekturen: Entwurf und Optimierung</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• J. L. Hennessy, D. A. Patterson, Computer Architecture: A Quantitative Approach, 2012</li> <li>• I. Koren, Computer Arithmetic Algorithms, 2001</li> <li>• Powerpoint Foliensatz</li> <li>• Auswahl von wissenschaftlichen Artikeln</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 569301 Vorlesung Grundlagen der Rechnerarchitektur</li> <li>• 569302 Übung Grundlagen der Rechnerarchitektur</li> </ul>		

16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h
	Selbststudiums- /	138 h
	Nachbearbeitungszeit:	
	Summe:	180 h
<hr/>		
17. Prüfungsnummer/n und -name:	56931 Grundlagen der Rechnerarchitektur (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0	
<hr/>		
18. Grundlage für ... :		
<hr/>		
19. Medienform:		
<hr/>		
20. Angeboten von:	Rechnerarchitektur	
<hr/>		

## Modul: 42410 Grundlagen des Wissenschaftlichen Rechnens

2. Modulkürzel:	051240020	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Dirk Pflüger		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Dirk Pflüger</li> <li>• Stefan Zimmer</li> <li>• Miriam Mehl</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 10190 Mathematik für Informatiker und Softwaretechniker und</li> <li>• Modul 10240 Numerische und Stochastische Grundlagen der Informatik bzw.</li> <li>• Modul 41590 Einführung in die Numerik und Stochastik für Softwaretechniker</li> </ul>		
12. Lernziele:	<p>Kenntnis grundlegender Konzepte, Algorithmen und Methoden des Wissenschaftlichen Rechnens. Fähigkeit, mit den erlernten Kenntnissen selbständig Methoden zu entwickeln, zu analysieren und umzusetzen, mit denen anwendungsorientierte Probleme effizient und genau gelöst werden können.</p>		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Überblick über die Simulationspipeline und die Wechselwirkungen zwischen den einzelnen Schritten</li> <li>• Skalenabhängige Modellierung</li> <li>• Diskretisierung (Gitter, Finite Elemente, Zeitschrittverfahren)</li> <li>• Algorithmen (Gittergenerierung, Adaptivität, Lineare Löser, Linked-Cell, Fast Multipole)</li> <li>• Parallelisierung (Gitterpartitionierung, Lastbalancierung)</li> <li>• Kurzer Überblick über die Visualisierung</li> </ul>		
14. Literatur:	<p>Martin Hanke-Bourgeois: Grundlagen der numerischen Mathematik und des wissenschaftlichen Rechnens; Vieweg+Teubner Verlag 2009</p>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 424101 Vorlesung Grundlagen des Wissenschaftlichen Rechnens</li> <li>• 424102 Übung Grundlagen des Wissenschaftlichen Rechnens</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	<p>42411 Grundlagen des Wissenschaftlichen Rechnens (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0</p>		

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von: Simulation großer Systeme

---

## Modul: 10170 Imaging Science

2. Modulkürzel:	051900210	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Andrés Bruhn		
9. Dozenten:	Andrés Bruhn		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 10190 Mathematik für Informatiker und Softwaretechniker</li> </ul>		
12. Lernziele:	<p>Der Student / die Studentin beherrscht die Grundlagen der Repräsentation und Verarbeitung digitaler Bilder, kann Probleme aus dem Fachgebiet einordnen und selbständig mit den erlernten Algorithmen und Verfahren lösen.</p> <p>The student knows the basics of digital image representation and processing and is able to solve problems of the field using the methods presented in the course.</p>		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Grundlagen aus der Optik:Lochkamera, Linsengleichung</li> <li>• Bildaufnahme:Kameras, Objektive, Beleuchtung, Aufnahmeprozess</li> <li>• Bildrepräsentation:Diskretisierung, Farbräume</li> <li>• Elementare Bildbearbeitung:Punktoperationen (z.B. Kontrastverstärkung, Binarisierung)</li> <li>• Lineare und nichtlineare Filter:Faltung, morphologische Operatoren</li> <li>• Fouriertransformation, Bilddarstellung und -bearbeitung im Fourierraum, Abtasttheorem</li> <li>• Orthogonale Transformationen:Cosinus, Wavelets</li> <li>• Kompression:Generische Verfahren (RLE, Entropie), spezielle Bildverfahren (z.B. jpeg)</li> <li>• Video:Formate, Kompression (z.B. MPEG)</li> <li>• Bildverbesserung und Restauration</li> <li>• Elementare Segmentierungsverfahren</li> </ul> <ul style="list-style-type: none"> <li>• Fundamentals of optics such as pinhole camera and lens equation</li> <li>• Image acquisition: Cameras, lenses, illumination, acquisition process</li> <li>• Image representation: Discretization, color spaces</li> <li>• Basics of image processing, e.g. point operations such as contrast enhancement or binarization</li> <li>• Linear and nonlinear filtering such as convolution and morphological operations.</li> <li>• Fourier transform, image representation and processing in Fourier space, sampling theorem</li> </ul>		

- Orthogonal transforms such as cosine transform and wavelets
  - Compression: Generic compression (RLE, entropy coding), methods specialized to domain of images (e.g. jpeg)
  - Video: file formats, compression (e.g. mpeg)
  - Image enhancement and restauration
  - Basics of segmentation
- 

14. Literatur:

- Bässmann, Henning; Kreys, Jutta: Bildverarbeitung Ad Oculos, 2004
  - Forsyth, David and Ponce, Jean: Computer Vision. A Modern Approach, 2003
  - Gonzalez, Rafael C.; Woods, Richard E.; Eddins, Steven L.: Digital Image Processing, 2004
  - Bigun, J.: Vision with Direction, 2006
  - Klaus D. Tönnies: Grundlagen der Bildverarbeitung, 2005
  - L. G. Shapiro, G. C. Stockman: Computer Vision, 2001
- 

15. Lehrveranstaltungen und -formen:

- 101701 Vorlesung Imaging Science
  - 101702 Übung Imaging Science
- 

16. Abschätzung Arbeitsaufwand:

Präsenzzeit:	42 h
Selbststudiums- /	138 h
Nachbearbeitungszeit:	
Summe:	180 h

---

17. Prüfungsnummer/n und -name:

- 10171 Imaging Science (PL), schriftlich, eventuell mündlich, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben.
  - V Vorleistung (USL-V), schriftlich, eventuell mündlich
- 

18. Grundlage für ... :

- 29430 Computer Vision
  - 55640 Correspondence Problems in Computer Vision
- 

19. Medienform:

---

20. Angeboten von:

Institut für Visualisierung und Interaktive Systeme

---

## Modul: 10180 Information Retrieval und Text Mining

2. Modulkürzel:	052401010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Roman Klinger		
9. Dozenten:	Roman Klinger		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 5. Semester          → Ergänzungsmodule --&gt;Katalog ISG          →</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester          → Ergänzungsmodule --&gt;Katalog ISW          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Katalog ISG          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Katalog ISW          →</p>		
11. Empfohlene Voraussetzungen:	Erfahrung mit Programmierung und Unix, erster Kontakt mit Verfahren des Maschinellen Lernens		
12. Lernziele:	Die Studierenden haben ein grundlegendes Verständnis der Konzepte und Algorithmen des Information Retrieval und Text Mining entwickelt.		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Textpräprozessierung</li> <li>• invertierte Indexe</li> <li>• IR-Modelle (z.B. Vektorraum-basiertes IR)</li> <li>• Linkanalyse</li> <li>• Clustering</li> <li>• Frage-Antwort-Systeme</li> <li>• korpusbasierter Erwerb von lexikalischem und Weltwissen</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Chris Manning, Prabhakar Raghavan, Hinrich Schütze, Introduction to Information Retrieval, 2008 Cambridge University Press</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 101801 Vorlesung Information Retrieval and Text Mining</li> <li>• 101802 Übung Information Retrieval and Text Mining</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 10181 Information Retrieval und Text Mining (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0</li> <li>• 10182 Information Retrieval und Text Mining - Hausübungen (USL), Sonstiges, Gewichtung: 1.0</li> </ul>		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Maschinelle Sprachverarbeitung		

## Modul: 10210 Mensch-Computer-Interaktion

2. Modulkürzel:	051900001	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Albrecht Schmidt		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Albrecht Schmidt</li> <li>• Thomas Ertl</li> <li>• Daniel Weiskopf</li> <li>• Niels Henze</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog SWT →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog SWT →</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 10280 Programmierung und Software-Entwicklung</li> </ul>		
12. Lernziele:	<p>Studierende entwickeln ein Verständnis für Modelle, Methoden und Konzepte der Mensch-Computer-Interaktion. Sie lernen verschiedene Ansätze für den Entwurf, die Entwicklung und Bewertung von Benutzungsschnittstellen kennen und verstehen deren Vor- und Nachteile.</p>		
13. Inhalt:	<p>Die Vorlesung vermittelt Konzepte, Prinzipien, Modelle, Methoden und Techniken für die effektive Entwicklung von benutzerfreundlichen Mensch-Computer-Schnittstellen. Das Thema moderner Benutzungsschnittstellen wird dabei für klassische Computer aber auch für mobile Geräte, eingebettete Systeme, Automobile und intelligente Umgebungen betrachtet.</p> <p>Die folgenden Themen werden in der Vorlesung behandelt:</p> <ul style="list-style-type: none"> <li>• Einführung in die Grundlagen der Mensch-Computer Interaktion, historische Entwicklung</li> <li>• Entwurfsprinzipien und Modelle für moderne Benutzungsschnittstellen und interaktive Systeme</li> <li>• Informationsverarbeitung des Menschen, Wahrnehmung, Motorik, Eigenschaften und Fähigkeiten des Benutzers</li> <li>• Interaktionskonzepte und -stile, Metaphern, Normen, Regeln und Style Guides</li> </ul>		

- Ein- und Ausgabegeräte, Entwurfsraum für interaktive Systeme
  - Analyse-, Entwurfs- und Entwicklungsmethoden und -werkzeuge für Benutzungsschnittstellen
  - Prototypische Realisierung und Implementierung von interaktiven Systemen, Werkzeuge
  - Architekturen für interaktive Systeme, User Interface Toolkits und Komponenten
  - Akzeptanz, Evaluationsmethoden und Qualitätssicherung
- 

14. Literatur:

- Bernhard Preim, Raimund Dachsel. Interaktive Systeme 1: Grundlagen, Graphical User Interfaces, Informationsvisualisierung. Springer, Berlin; 2. Auflage. 2010
  - Alan Dix, Janet Finley, Gregory Abowd, Russell Beale, Human-Computer Interaction, 2004
  - Ben Shneiderman, Catherine Plaisant, Designing the User Interfaces, 2005
- 

15. Lehrveranstaltungen und -formen:

- 102101 Vorlesung Mensch-Computer-Interaktion
  - 102102 Übung Mensch-Computer-Interaktion
- 

16. Abschätzung Arbeitsaufwand:

Präsenzzeit:	42 h
Selbststudiums- /	138 h
Nachbearbeitungszeit:	
Summe:	180 h

---

17. Prüfungsnummer/n und -name:

- 10211 Mensch-Computer-Interaktion (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein
  - V Vorleistung (USL-V), schriftlich, eventuell mündlich
- 

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von:

Institut für Visualisierung und Interaktive Systeme

---

## Modul: 10220 Modellierung

2. Modulkürzel:	052010001	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Bernhard Mitschang</li> <li>• Frank Leymann</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog SWT →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog SWT →</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• 051520005 Programmierung und Software-Entwicklung</li> <li>• 051510005 Datenstrukturen und Algorithmen</li> <li>• 051200005 Systemkonzepte und -programmierung</li> </ul>		
12. Lernziele:	<p>Am Ende des Moduls sind die Studierenden in der Lage, wesentliche Artefakte eines IT Systems zu modellieren. Der Zusammenhang und das Zusammenspiel solcher Artefakte ist verstanden. Die Rolle von Metamodellen und deren Erstellung ist klar.</p>		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Entity-Relationship Modell &amp; komplexe Objekte</li> <li>• Relationenmodell &amp; Relationenalgebra , Überblick SQL</li> <li>• Transformationen von ER nach Relationen, Normalisierung</li> <li>• XML, DTD, XML-Schema, Info-Set, Namensräume</li> <li>• Metamodelle &amp; Repository</li> <li>• RDF, RDF-S &amp; Ontologien</li> <li>• UML</li> <li>• Petri Netze, Workflownetze</li> <li>• BPMN</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• A. Silberschatz, H. F. Korth, S. Sudarshan, Database System Concepts, 2002</li> <li>• R. Eckstein, S. Eckstein, "XML und Datenmodellierung", dpunkt.verlag 2004</li> <li>• M. Hitz, G. Kappel, E. Kapsammer, W. Retschitzegger, UML @ Work - Objektorientierte Modellierung mit UML2, 2005</li> <li>• P. Hitzler, M. Krötzsch, S. Rudolph, Y. Sure, Semantic Web, 2008</li> <li>• T.J. Teorey, Database Modeling &amp; Design, 2nd Edition, 1994</li> </ul>		

- H.J. Habermann, F. Leymann, "Repository", Oldenbourg 1993
- W. Reisig, "Petri-Netze", Vieweg & Teubner 2010
- B. Silver, "BPMN Method & Style", Cody-Cassidy Press 2009

---

15. Lehrveranstaltungen und -formen:

- 102201 Vorlesung Modellierung
- 102202 Übung Modellierung

---

16. Abschätzung Arbeitsaufwand:

Präsenzzeit: 42 Stunden  
Nachbearbeitungszeit: 138 Stunden

---

17. Prüfungsnummer/n und -name:

- 10221 Modellierung (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein
- V Vorleistung (USL-V), schriftlich, eventuell mündlich

---

18. Grundlage für ... :

- 10030 Architektur von Anwendungssystemen
- 10080 Datenbanken und Informationssysteme

---

19. Medienform:

---

20. Angeboten von:

Institut für Architektur von Anwendungssystemen

---

## Modul: 39040 Rechnernetze

2. Modulkürzel:	051200010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Kurt Rothermel		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Kurt Rothermel</li> <li>• Frank Dürr</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• 051520005 Programmierung und Software-Entwicklung</li> <li>• 051510005 Datenstrukturen und Algorithmen</li> <li>• Grundkenntnisse in Java</li> </ul>		
12. Lernziele:	<ul style="list-style-type: none"> <li>• Versteht grundsätzliche Eigenschaften, Konzepte und Methoden von Rechnernetzen, insbesondere dem Internet.</li> <li>• Versteht Schichten und deren Zusammenwirken in einem Protokollstapel</li> <li>• Kann Rechnernetze aufbauen, verwalten und analysieren.</li> <li>• Kann Protokolle entwickeln und in Schichtenarchitektur einbetten.</li> <li>• Kann höhere Kommunikationsdienste zur Entwicklung von netzgestützten Systemen anwenden.</li> <li>• Kann sich mit Experten anderer Domänen über Methoden der Rechnernetze verständigen.</li> </ul>		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Einführung in die Rechnernetze, ISO Referenzmodell;</li> <li>• Bitübertragungsschicht: Übertragungsmedien, analoge und digitale Informationskodierung und -übertragung, Vermittlungsarten;</li> <li>• Sicherungsschicht: Betriebsarten, Fehlererkennung und -behandlung, Flusskontrolle;</li> <li>• Lokale Netze: CSMA/CD, Token Ring, Token Bus, FDDI, Kopplung;</li> <li>• Vermittlungsschicht: Verbindungsorientierter und verbindungsloser Dienst, Leitwegbestimmung, Überlastkontrolle;</li> <li>• Internetworking;</li> <li>• Internet-Protokoll;</li> <li>• Transportschicht: ausgewählte Realisierungsprobleme und Internet-Protokolle;</li> <li>• Echtzeitkommunikation: IntServ, DiffServ; Sicherheit: Verfahren, IPsec, SSL, TLS.</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• A.S. Tanenbaum, Computer Networks, 4th Edition, 2003</li> <li>• D.E. Comer, Computernetzwerke und Internets, 2000</li> <li>• D.E. Comer, Internetworking with TCP/IP Volume I: Principles, Protocols, and Architecture, 1995</li> </ul>		

- J. F. Kurose, K. W. Ross, Computer Networks: a top-down approach featuring the Internet, 2001
  - L.L. Peterson, B.S. Davie, Computer Networks: A Systems Approach, 1999
- 

15. Lehrveranstaltungen und -formen:

- 390401 VL Rechnernetze
- 390402 ÜB Rechnernetze

---

16. Abschätzung Arbeitsaufwand:

Präsenzzeit: 42 Stunden  
Nachbearbeitungszeit: 138 Stunden

---

17. Prüfungsnummer/n und -name:

- 39041 Rechnernetze (PL), schriftlich oder mündlich, 90 Min.,  
Gewichtung: 1.0, Prüfungsdauer: 90 min schriftlich oder 30 min mündlich Exam duration: 90 min written exam or 30 min oral exam
- V Vorleistung (USL-V), schriftlich, eventuell mündlich

---

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von: Verteilte Systeme

---

## Modul: 36530 Rechnerorganisation 1

2. Modulkürzel:	051700006	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Hans-Joachim Wunderlich		
9. Dozenten:	Hans-Joachim Wunderlich		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog SWT →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog SWT →</p>		
11. Empfohlene Voraussetzungen:	Einführung in die Technische Informatik (14360)		
12. Lernziele:	<ul style="list-style-type: none"> <li>• Kenntnisse über die Grundlagen der Organisation von Rechnern und deren maschinennahe Programmierung</li> <li>• Grundzüge über die Beschreibung und den Entwurf von Hardwaresystemen</li> </ul>		
13. Inhalt:	<p>Aufbau, Entwurf und maschinennahe Programmierung von Mikroprozessoren werden erläutert. In den Übungen wird das Wissen durch schriftliche Aufgaben sowie Experimente mit Prozessorsimulatoren vertieft. Im einzelnen werden behandelt:</p> <ul style="list-style-type: none"> <li>• Grundstrukturen: Stack-, Akkumulator- und Register-basierende Rechner</li> <li>• Informationsdarstellung in Hardware: Daten- und Befehlsformate, Fehlererkennung und -korrektur, Befehlssätze und Unterstützung von Hochsprachen.</li> <li>• MIPS als RISC-Bespiel und seine Assemblerprogrammierung</li> <li>• Grundelemente und Entwurf auf Register-Transfer-Ebene</li> <li>• Grundzüge einer Hardware-Beschreibungssprache</li> <li>• Operationswerke: Multiplikation, Division, Gleitkommaeinheiten</li> <li>• Steuerwerksentwurf und Mikroprogrammierung</li> <li>• Befehlszyklus und Unterbrechungen</li> <li>• Pipelining und statisches Scheduling</li> <li>• Speicherorganisation: Cachestrukturen und virtueller Speicher, Seitenverwaltung, Segmentierung, TLB, MMU und DMA</li> <li>• Leistungsbewertung: Maßzahlen und CPI, Benchmarking und einfache Anwendung von Warteschlangen</li> </ul>		

14. Literatur:	Literatur, siehe Webseite zur Veranstaltung								
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"><li>• 365301 Vorlesung Rechnerorganisation 1</li><li>• 365302 Praktische und theoretische Übungen Rechnerorganisation 1</li></ul>								
16. Abschätzung Arbeitsaufwand:	<table><tr><td>Präsenzzeit:</td><td>42 h</td></tr><tr><td>Selbststudiums- /</td><td>138 h</td></tr><tr><td>Nachbearbeitungszeit:</td><td></td></tr><tr><td>Summe:</td><td>180 h</td></tr></table>	Präsenzzeit:	42 h	Selbststudiums- /	138 h	Nachbearbeitungszeit:		Summe:	180 h
Präsenzzeit:	42 h								
Selbststudiums- /	138 h								
Nachbearbeitungszeit:									
Summe:	180 h								
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"><li>• 36531 Rechnerorganisation 1 (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0</li><li>• V Vorleistung (USL-V), schriftliche Prüfung</li></ul>								
18. Grundlage für ... :									
19. Medienform:									
20. Angeboten von:	Rechnerarchitektur								

---

## Modul: 40090 Systemkonzepte und -programmierung

2. Modulkürzel:	051200005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Kurt Rothermel		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Kurt Rothermel</li> <li>• Frank Dürr</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 5. Semester          → Ergänzungsmodule --&gt;Katalog ISG          →</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester          → Ergänzungsmodule --&gt;Katalog ISW          →</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester          → Ergänzungsmodule --&gt;Katalog SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Katalog ISG          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Katalog ISW          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Katalog SWT          →</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>* Modul 051520005 Programmierung und Software-Entwicklung</li> <li>* Modul 051510005 Datenstrukturen und Algorithmen</li> </ul>		
12. Lernziele:	<ul style="list-style-type: none"> <li>* Verstehen grundlegender Architekturen und Organisationsformen von Software-Systemen</li> <li>* Verstehen systemnaher Konzepte und Mechanismen</li> <li>* Kann existierende Systemplattformen und Betriebssysteme hinsichtlich ihrer Eigenschaften analysieren und anwenden.</li> <li>* Kann systemnahe Software entwerfen und implementieren.</li> <li>* Kann nebenläufige Programme entwickeln</li> <li>* Kann mit Experten anderer Fachgebiete die Anwendung von Systemfunktionen abstimmen.</li> </ul>		
13. Inhalt:	<p>Grundlegende Systemstrukturen - und organisationen</p> <ul style="list-style-type: none"> <li>• Multitaskingsystem</li> <li>• Multiprozessorsystem</li> <li>• Verteiltes System</li> </ul> <p>Modellierung und Analyse nebenläufiger Programme</p> <ul style="list-style-type: none"> <li>• Abstraktionen: Atomare Befehle, Prozesse, nebenläufiges Programm</li> <li>• Korrektheit- und Leitungskriterien</li> </ul> <p>Betriebssystemkonzepte</p> <ul style="list-style-type: none"> <li>• Organisation von Betriebssystemen</li> <li>• Prozesse und Threads</li> <li>• Eingabe/Ausgabe</li> <li>• Scheduling</li> </ul>		

Konzepte zur Synchronisation über gemeinsamen Speicher

- Synchronisationsprobleme und -lösungen
- Synchronisationswerkzeuge: Semaphor, Monitor

Konzepte zur Kommunikation und Synchronisation mittels Nachrichtentransfer

- Taxonomie: Kommunikation und Synchronisation
- Nachrichten als Kommunikationskonzept
- Höhere Kommunikationskonzepte

Basialgorithmen für Verteilte Systeme

- Erkennung globaler Eigenschaften
- Schnappschussproblem
- Konsistenter globaler Zustand
- Verteilte Terminierung

Praktische nebenläufige Programmierung in Java

- Threads und Synchronisation
- Socketschnittstelle
- RMI Programmierung

---

14. Literatur:	Literatur, siehe Webseite zur Veranstaltung
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"><li>• 400901 Vorlesung Systemkonzepte und -programmierung</li><li>• 400902 Übung Systemkonzepte und -programmierung</li></ul>
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"><li>• 40091 Systemkonzepte und -programmierung (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0</li><li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich</li></ul>
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Verteilte Systeme

---

## 330 Katalog ISW

---

Zugeordnete Module:	10030	Architektur von Anwendungssystemen
	10060	Computergraphik
	10100	Grundlagen der Graphischen Ingenieursysteme
	10110	Grundlagen der Künstlichen Intelligenz
	10170	Imaging Science
	10180	Information Retrieval und Text Mining
	10210	Mensch-Computer-Interaktion
	10220	Modellierung
	18560	Grundlagen der Eingebetteten Systeme
	36530	Rechnerorganisation 1
	39040	Rechnernetze
	40090	Systemkonzepte und -programmierung
	41590	Einführung in die Numerik und Stochastik für Softwaretechniker
	42410	Grundlagen des Wissenschaftlichen Rechnens
	56930	Grundlagen der Rechnerarchitektur

---

## Modul: 10030 Architektur von Anwendungssystemen

2. Modulkürzel:	052010002	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	Frank Leymann		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p>		
11. Empfohlene Voraussetzungen:	Vorlesungen des Grundstudiums.		
12. Lernziele:	<p>Die Vorlesung erläutert den Begriff der Architektur von Anwendungssystemen und die Rolle des Architekten solcher Systeme. Die wesentlichen Bestandteile von Anwendungsarchitektur wie etwa Datenbanksysteme, Anwendungsserver, Messaging Systeme, Workflowsysteme und TP-Monitore werden diskutiert. Die wesentlichen Mustern zur Erstellung von Anwendungssystemen sind verstanden.</p>		
13. Inhalt:	<p>Architekturelle Stile wie etwa N-stufige Aufbauten oder Service-Orientierung werden vorgestellt. Architekturmuster werden detailliert. Fundamentale Konzepte wie Transaktionen und Queuing werden eingeführt. Darauf aufbauend wird Direct TP vs Queues TP diskutiert. Grundlegende Qualitätseigenschaften wie Verfügbarkeit und Skalierbarkeit werden erläutert und Mechanismen zu deren Erzielen eingeführt. Die Rolle von Komponenten und Programmierung im Großen wird heraus gearbeitet und Modell-getriebene Architektur vorgestellt.</p>		
14. Literatur:	<ul style="list-style-type: none"> <li>• A. Silberschatz, H. F. Korth, S. Sudarshan, Database System Concepts, 2002</li> <li>• B. Neubauer, T. Ritter, F. Stoinnski, CORBA Komponenten, 2004</li> <li>• F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal, Pattern-orientierte Software Architektur - Ein Patternsystem, 1998</li> <li>• F. Leymann, D. Roller, Production Workflow, 2000</li> <li>• L. Hohmann, Beyond Software Architecture, 2003</li> <li>• M. Fowler, Patters of Enterprise Application Architecture, 2003</li> <li>• P. Bernstein, E. Newcomer, Principles of Transaction Processing, 1997</li> <li>• S. Conrad, W. Hasselbring, A. Koschel, R. Tritsch, Enterprise Application Integration, 2006</li> <li>• S. Weerawarana, F. Curbera, F. Leymann, T. Storey, D. Ferguson, Web Services Platform Architecture, 2005</li> </ul>		

	<ul style="list-style-type: none"><li>• W. Emmerich, Konstruktion von verteilten Objekten, 2003</li></ul>
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"><li>• 100301 Vorlesung Grundlagen der Architektur von Anwendungssystemen</li><li>• 100302 Übung Grundlagen der Architektur von Anwendungssystemen</li></ul>
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"><li>• 10031 Architektur von Anwendungssystemen (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0,</li><li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich</li></ul>
18. Grundlage für ... :	<ul style="list-style-type: none"><li>• 29480 Loose Coupling and Message Based Applications</li><li>• 29510 Service Computing</li><li>• 29530 Business Process Management</li><li>• 42520 Services and Service Composition</li></ul>
19. Medienform:	Vorlesungen mit begleitenden Übungen
20. Angeboten von:	Architektur von Anwendungssystemen

---

## Modul: 10060 Computergraphik

2. Modulkürzel:	051900002	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Thomas Ertl		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Thomas Ertl</li> <li>• Daniel Weiskopf</li> <li>• Martin Fuchs</li> <li>• Guido Reina</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 10210 Mensch-Computer-Interaktion</li> <li>• Modul 41590 Einführung in die Numerik und Stochastik</li> </ul>		
12. Lernziele:	Die Studierenden haben Wissen über die Grundlagen der Computergraphik sowie praktische Fähigkeiten in der Graphikprogrammierung erworben.		
13. Inhalt:	<p>Folgende Themen werden in der Vorlesung behandelt:</p> <ul style="list-style-type: none"> <li>• Überblick über den Prozess der Bildsynthese</li> <li>• Graphische Geräte, visuelle Wahrnehmung, Farbsysteme</li> <li>• Grundlegende Rastergraphik und Bildverarbeitung</li> <li>• Raytracing und Beleuchtungsmodelle</li> <li>• 2D und 3D Geometrietransformationen, 3D Projektion</li> <li>• Graphikprogrammierung in OpenGL 3</li> <li>• Texturen</li> <li>• Polygonale und hierarchische Modelle</li> <li>• Rasterisierung und Verdeckungsrechnung</li> <li>• Grundlagen der geometrischen Modellierung (Kurven, Flächen)</li> <li>• Räumliche Datenstrukturen</li> </ul> <p>Die Veranstaltung besteht aus Vorlesung mit Übungen. Die Übungen umfassen praktische Programmierübungen, theoretische Themen und Programmierprojekte.</p>		
14. Literatur:	<ul style="list-style-type: none"> <li>• J. Encarnacao, W. Strasser, R. Klein: Graphische Datenverarbeitung (Band1 und 2), 1997</li> <li>• J. Foley, A. van Dam, S. Feiner, J. Hughes: Computer Graphics: Principle and Practice, 1990</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 100601 Vorlesung Computergraphik</li> <li>• 100602 Übung Computergraphik</li> </ul>		

16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h
	Selbststudiums- /	138 h
	Nachbearbeitungszeit:	
	Summe:	180 h
<hr/>		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"><li>• 10061 Computergraphik (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein.</li><li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich</li></ul>	
<hr/>		
18. Grundlage für ... :		
<hr/>		
19. Medienform:		
<hr/>		
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme	
<hr/>		

## Modul: 41590 Einführung in die Numerik und Stochastik für Softwaretechniker

2. Modulkürzel:	051240006	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Dirk Pflüger		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Dirk Pflüger</li> <li>• Stefan Zimmer</li> <li>• Thomas Ertl</li> <li>• Miriam Mehl</li> <li>• Daniel Weiskopf</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 5. Semester          → Ergänzungsmodule --&gt;Katalog ISG          →</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester          → Ergänzungsmodule --&gt;Katalog ISW          →</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester          → Ergänzungsmodule --&gt;Katalog SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Katalog ISG          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Katalog ISW          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Katalog SWT          →</p>		
11. Empfohlene Voraussetzungen:	Mathematik für Informatiker und Softwaretechniker (Modulnummer 10190)		
12. Lernziele:	Beherrschung grundlegender Begriffe und Methoden der Numerik und Stochastik, Kenntnis der Anwendungsbereiche und Gültigkeitsgrenzen der erlernten Methoden, insbesondere Kenntnis der Auswirkungen von Näherungen, Beherrschung der Modellierung einfacher Probleme mit stochastischen Methoden.		
13. Inhalt:	<p>Methoden der angewandten Mathematik, insbesondere der Numerik und Stochastik, sind für viele Bereiche der Informatik wie Simulation, Grafik oder Bildverarbeitung von zentraler Bedeutung. In Ergänzung der Mathematik-Grundausbildung vermittelt diese Vorlesung folgende Grundkenntnisse:</p> <ul style="list-style-type: none"> <li>• numerische Algorithmik</li> <li>• Gleitpunktzahlen und Gleitpunktarithmetik</li> <li>• Interpolation &amp; Approximation</li> <li>• Integration</li> <li>• lineare Gleichungssysteme</li> <li>• Iterative Lösung linearer und nichtlinearer Gleichungen</li> <li>• gewöhnliche Differentialgleichungen</li> <li>• Stochastik</li> </ul>		

- Zufall und Unsicherheit
  - diskrete und kontinuierliche Wahrscheinlichkeitsräume
  - Asymptotik
- 

14. Literatur:

- Dahmen, Reusken: Numerik für Ingenieure
  - Schwarz, Köckler: Numerische Mathematik
  - Huckle, Schneider: Numerik für Informatiker
  - Henze: Stochastik für Einsteiger
  - Schickinger, Steger: Diskrete Strukturen, Band 2
- 

15. Lehrveranstaltungen und -formen:

- 415901 Vorlesung Einführung in die Numerik und Stochastik für Softwaretechniker
  - 415902 Übung Einführung in die Numerik und Stochastik für Softwaretechniker
- 

16. Abschätzung Arbeitsaufwand:

Präsenzzeit:	42 h
Selbststudiums- /	138 h
Nachbearbeitungszeit:	
Summe:	180 h

---

17. Prüfungsnummer/n und -name:

41591 Einführung in die Numerik und Stochastik für Softwaretechniker (PL), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 1.0

---

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von:

Simulation großer Systeme

---

## Modul: 18560 Grundlagen der Eingebetteten Systeme

2. Modulkürzel:	051711010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Martin Radetzki		
9. Dozenten:	Martin Radetzki		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISG → B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISW →		
11. Empfohlene Voraussetzungen:	Modul 051700005 Rechnerorganisation		
12. Lernziele:	Methodische und werkzeugunterstützte Konstruktion von Hardware und Software für auf spezifische Anwendungen hin optimierte eingebettete Systeme.		
13. Inhalt:	Einführung in Charakteristik und Anwendungsgebiete eingebetteter Systeme, hardwarenahe Softwareentwicklung, Software-Scheduling, eingebettete Algorithmen (digitale Signalverarbeitung, Kanal- und Quellencodierung am Beispiel Viterbi-Algorithmus und MPEG-Video-Codec), zustandsbasierte Modellierung (Statecharts), eingebettete Prozessoren (Microcontroller, digitale Signalprozessoren, ARM), Bussysteme und Speicher, Systemsynthese (Taskgraphen, Allokation, Bindung, Ablaufplanung/Scheduling)		
14. Literatur:	J. Teich, Digitale Hardware/Software-Systeme, 2. Auflage, 2007 P. Marwedel, Embedded System Design, 2006		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 185601 Vorlesung Grundlagen der Eingebetteten Systeme</li> <li>• 185602 Übung Grundlagen der Eingebetteten Systeme</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 18561 Grundlagen der Eingebetteten Systeme (PL), schriftliche Prüfung, 90 Min., Gewichtung: 75.0</li> <li>• V Vorleistung (USL-V), schriftliche Prüfung</li> </ul>		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Eingebettete Systeme (Embedded Systems Engineering)		

## Modul: 10100 Grundlagen der Graphischen Ingenieursysteme

2. Modulkürzel:	051400005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Dieter Roller		
9. Dozenten:	Dieter Roller		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p>		
11. Empfohlene Voraussetzungen:	Grundstudium		
12. Lernziele:	<ul style="list-style-type: none"> <li>• Kenntnis und Verständnis von Modellen bei der Produktentwicklung</li> <li>• Grundkenntnisse über die wichtigsten Modellarten, Algorithmen und Datenstrukturen und Techniken für den Datenaustausch</li> </ul>		
13. Inhalt:	<p>Inhalte:</p> <ul style="list-style-type: none"> <li>• Anforderungen an CAD-Systeme</li> <li>• zweidimensionale Modelle</li> <li>• dreidimensionale Modelle</li> <li>• interaktive Modellerstellung</li> <li>• Einführung in die Modifikationstechnik u. parametrische Modellierung</li> <li>• Methoden zur Modellmodifikation</li> <li>• Grundlagen der parametrischen Modellierung</li> <li>• Ansätze und Verfahren zur parametrischen Variantenerzeugung</li> <li>• Ausgewählte Anwendungsbeispiele</li> <li>• Überblick über weitergehende Modellieransätze</li> <li>• Datenverwaltung in CAD</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• D. Roller, CAD - Effiziente Anpassungs- und Variantenkonstruktion, Springer-Verlag</li> <li>• Literatur, siehe Webseite zur Veranstaltung</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 101001 Vorlesung Grundlagen der Graphischen Ingenieursysteme</li> <li>• 101002 Übung Grundlagen der Graphischen Ingenieursysteme</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	10101 Grundlagen der Graphischen Ingenieursysteme (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0		
18. Grundlage für ... :			

19. Medienform:

20. Angeboten von:

Institut für Rechnergestützte Ingenieursysteme

---

## Modul: 10110 Grundlagen der Künstlichen Intelligenz

2. Modulkürzel:	051900205	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Marc Toussaint		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Andrés Bruhn</li> <li>• Marc Toussaint</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 080300100 Mathematik für Informatiker und Softwaretechniker</li> </ul>		
12. Lernziele:	Der Student / die Studentin beherrscht die Grundlagen der Künstlichen Intelligenz, kann Probleme der KI selbständig einordnen und mit den erlernten Methoden und Algorithmen bearbeiten.		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Intelligenz</li> <li>• Agentenbegriff</li> <li>• Problemlösen durch Suchen, Suchverfahren</li> <li>• Probleme mit Rand- und Nebenbedingungen</li> <li>• Spiele</li> <li>• Aussagen- und Prädikatenlogik</li> <li>• Logikbasierte Agenten, Wissensrepräsentation</li> <li>• Inferenz</li> <li>• Planen</li> <li>• Unsicherheit, probabilistisches Schließen</li> <li>• Probabilistisches Schließen über die Zeit</li> <li>• Sprachverarbeitung</li> <li>• Entscheidungstheorie</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• S. Russell, P. Norvig, Künstliche Intelligenz, 2004</li> <li>• G. F. Luger, Künstliche Intelligenz, 2001</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 101101 Vorlesung Grundlagen der Künstlichen Intelligenz</li> <li>• 101102 Übung Grundlagen der Künstlichen Intelligenz</li> </ul>		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden</p> <p><b>Gesamt: 180 Stunden</b></p>		

17. Prüfungsnummer/n und -name:
- 10111 Grundlagen der Künstlichen Intelligenz (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben
  - V Vorleistung (USL-V), schriftlich, eventuell mündlich
- 

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von: Institut für Visualisierung und Interaktive Systeme

---

## Modul: 56930 Grundlagen der Rechnerarchitektur

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Hans-Joachim Wunderlich		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Michael Kochte</li> <li>• Hans-Joachim Wunderlich</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 41930 Rechnerorganisation</li> </ul>		
12. Lernziele:	<ul style="list-style-type: none"> <li>• Tiefes Verständnis von Entwurfskonzepten, die in modernen Prozessoren und Rechensystemen Verwendung finden</li> <li>• Kenntnis von Entwurfsherausforderungen.</li> <li>• Verständnis von aktuellen und zukünftigen Entwicklungstrends</li> </ul>		
13. Inhalt:	<p>Grundlegende und fortgeschrittene Themen der Rechnerarchitektur, inklusive:</p> <ul style="list-style-type: none"> <li>• Technologiegrundlagen: Entwurfsverfahren, Herstellungsmethoden, Ausbeute, Test und Zuverlässigkeit, Kosten und Qualität, Skalierung.</li> <li>• Rechen- und Verlustleistung: Analyse und Optimierung</li> <li>• Rechnerarithmetik: Effiziente Hardwarestrukturen für grundlegende Arithmetik, Implementierung von Logarithmen, Exponentialfunktion und trigonometrischen Funktionen, arithmetische Pipelines, praktische Implementierungen von Gleitkommaarithmetik (Cell SPE, SPARC).</li> <li>• Instruktionsparallelismus (ILP): Superskalarität, statisches und dynamisches Scheduling, out-of-order execution, VLIW Prozessoren, Multithreading.</li> <li>• Datenparallelismus (DLP): Vektorprozessoren, SIMD, Grafikprozessoreinheiten (GPGPU)</li> <li>• Threadparallelismus (TLP): Mehrprozessorsysteme, Speicherkohärenz und Synchronisierung.</li> <li>• Speicher- und Cache-Architekturen: Entwurf und Optimierung</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• J. L. Hennessy, D. A. Patterson, Computer Architecture: A Quantitative Approach, 2012</li> <li>• I. Koren, Computer Arithmetic Algorithms, 2001</li> <li>• Powerpoint Foliensatz</li> <li>• Auswahl von wissenschaftlichen Artikeln</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 569301 Vorlesung Grundlagen der Rechnerarchitektur</li> <li>• 569302 Übung Grundlagen der Rechnerarchitektur</li> </ul>		

16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h
	Selbststudiums- /	138 h
	Nachbearbeitungszeit:	
	Summe:	180 h
<hr/>		
17. Prüfungsnummer/n und -name:	56931 Grundlagen der Rechnerarchitektur (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0	
<hr/>		
18. Grundlage für ... :		
<hr/>		
19. Medienform:		
<hr/>		
20. Angeboten von:	Rechnerarchitektur	
<hr/>		

## Modul: 42410 Grundlagen des Wissenschaftlichen Rechnens

2. Modulkürzel:	051240020	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Dirk Pflüger		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Dirk Pflüger</li> <li>• Stefan Zimmer</li> <li>• Miriam Mehl</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 10190 Mathematik für Informatiker und Softwaretechniker und</li> <li>• Modul 10240 Numerische und Stochastische Grundlagen der Informatik bzw.</li> <li>• Modul 41590 Einführung in die Numerik und Stochastik für Softwaretechniker</li> </ul>		
12. Lernziele:	<p>Kenntnis grundlegender Konzepte, Algorithmen und Methoden des Wissenschaftlichen Rechnens. Fähigkeit, mit den erlernten Kenntnissen selbständig Methoden zu entwickeln, zu analysieren und umzusetzen, mit denen anwendungsorientierte Probleme effizient und genau gelöst werden können.</p>		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Überblick über die Simulationspipeline und die Wechselwirkungen zwischen den einzelnen Schritten</li> <li>• Skalenabhängige Modellierung</li> <li>• Diskretisierung (Gitter, Finite Elemente, Zeitschrittverfahren)</li> <li>• Algorithmen (Gittergenerierung, Adaptivität, Lineare Löser, Linked-Cell, Fast Multipole)</li> <li>• Parallelisierung (Gitterpartitionierung, Lastbalancierung)</li> <li>• Kurzer Überblick über die Visualisierung</li> </ul>		
14. Literatur:	<p>Martin Hanke-Bourgeois: Grundlagen der numerischen Mathematik und des wissenschaftlichen Rechnens; Vieweg+Teubner Verlag 2009</p>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 424101 Vorlesung Grundlagen des Wissenschaftlichen Rechnens</li> <li>• 424102 Übung Grundlagen des Wissenschaftlichen Rechnens</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	<p>42411 Grundlagen des Wissenschaftlichen Rechnens (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0</p>		

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von: Simulation großer Systeme

---

## Modul: 10170 Imaging Science

2. Modulkürzel:	051900210	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:		Univ.-Prof. Andrés Bruhn	
9. Dozenten:		Andrés Bruhn	
10. Zuordnung zum Curriculum in diesem Studiengang:		<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p>	
11. Empfohlene Voraussetzungen:		<ul style="list-style-type: none"> <li>• Modul 10190 Mathematik für Informatiker und Softwaretechniker</li> </ul>	
12. Lernziele:		<p>Der Student / die Studentin beherrscht die Grundlagen der Repräsentation und Verarbeitung digitaler Bilder, kann Probleme aus dem Fachgebiet einordnen und selbständig mit den erlernten Algorithmen und Verfahren lösen.</p> <p>The student knows the basics of digital image representation and processing and is able to solve problems of the field using the methods presented in the course.</p>	
13. Inhalt:		<ul style="list-style-type: none"> <li>• Grundlagen aus der Optik:Lochkamera, Linsengleichung</li> <li>• Bildaufnahme:Kameras, Objektive, Beleuchtung, Aufnahmeprozess</li> <li>• Bildrepräsentation:Diskretisierung, Farbräume</li> <li>• Elementare Bildbearbeitung:Punktoperationen (z.B. Kontrastverstärkung, Binarisierung)</li> <li>• Lineare und nichtlineare Filter:Faltung, morphologische Operatoren</li> <li>• Fouriertransformation, Bilddarstellung und -bearbeitung im Fourierraum, Abtasttheorem</li> <li>• Orthogonale Transformationen:Cosinus, Wavelets</li> <li>• Kompression:Generische Verfahren (RLE, Entropie), spezielle Bildverfahren (z.B. jpeg)</li> <li>• Video:Formate, Kompression (z.B. MPEG)</li> <li>• Bildverbesserung und Restauration</li> <li>• Elementare Segmentierungsverfahren</li> </ul> <ul style="list-style-type: none"> <li>• Fundamentals of optics such as pinhole camera and lens equation</li> <li>• Image acquisition: Cameras, lenses, illumination, acquisition process</li> <li>• Image representation: Discretization, color spaces</li> <li>• Basics of image processing, e.g. point operations such as contrast enhancement or binarization</li> <li>• Linear and nonlinear filtering such as convolution and morphological operations.</li> <li>• Fourier transform, image representation and processing in Fourier space, sampling theorem</li> </ul>	

- Orthogonal transforms such as cosine transform and wavelets
  - Compression: Generic compression (RLE, entropy coding), methods specialized to domain of images (e.g. jpeg)
  - Video: file formats, compression (e.g. mpeg)
  - Image enhancement and restauration
  - Basics of segmentation
- 

14. Literatur:
- Bässmann, Henning; Kreys, Jutta: Bildverarbeitung Ad Oculos, 2004
  - Forsyth, David and Ponce, Jean: Computer Vision. A Modern Approach, 2003
  - Gonzalez, Rafael C.; Woods, Richard E.; Eddins, Steven L.: Digital Image Processing, 2004
  - Bigun, J.: Vision with Direction, 2006
  - Klaus D. Tönnies: Grundlagen der Bildverarbeitung, 2005
  - L. G. Shapiro, G. C. Stockman: Computer Vision, 2001
- 

15. Lehrveranstaltungen und -formen:
- 101701 Vorlesung Imaging Science
  - 101702 Übung Imaging Science
- 

16. Abschätzung Arbeitsaufwand:
- |                       |       |
|-----------------------|-------|
| Präsenzzeit:          | 42 h  |
| Selbststudiums- /     | 138 h |
| Nachbearbeitungszeit: |       |
| Summe:                | 180 h |
- 

17. Prüfungsnummer/n und -name:
- 10171 Imaging Science (PL), schriftlich, eventuell mündlich, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben.
  - V Vorleistung (USL-V), schriftlich, eventuell mündlich
- 

18. Grundlage für ... :
- 29430 Computer Vision
  - 55640 Correspondence Problems in Computer Vision
- 

19. Medienform:

---

20. Angeboten von: Institut für Visualisierung und Interaktive Systeme

---

## Modul: 10180 Information Retrieval und Text Mining

2. Modulkürzel:	052401010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Roman Klinger		
9. Dozenten:	Roman Klinger		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p>		
11. Empfohlene Voraussetzungen:	Erfahrung mit Programmierung und Unix, erster Kontakt mit Verfahren des Maschinellen Lernens		
12. Lernziele:	Die Studierenden haben ein grundlegendes Verständnis der Konzepte und Algorithmen des Information Retrieval und Text Mining entwickelt.		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Textpräprozessierung</li> <li>• invertierte Indexe</li> <li>• IR-Modelle (z.B. Vektorraum-basiertes IR)</li> <li>• Linkanalyse</li> <li>• Clustering</li> <li>• Frage-Antwort-Systeme</li> <li>• korpusbasierter Erwerb von lexikalischem und Weltwissen</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Chris Manning, Prabhakar Raghavan, Hinrich Schütze, Introduction to Information Retrieval, 2008 Cambridge University Press</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 101801 Vorlesung Information Retrieval and Text Mining</li> <li>• 101802 Übung Information Retrieval and Text Mining</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 10181 Information Retrieval und Text Mining (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0</li> <li>• 10182 Information Retrieval und Text Mining - Hausübungen (USL), Sonstiges, Gewichtung: 1.0</li> </ul>		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Maschinelle Sprachverarbeitung		

## Modul: 10210 Mensch-Computer-Interaktion

2. Modulkürzel:	051900001	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Albrecht Schmidt		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Albrecht Schmidt</li> <li>• Thomas Ertl</li> <li>• Daniel Weiskopf</li> <li>• Niels Henze</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog SWT →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog SWT →</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 10280 Programmierung und Software-Entwicklung</li> </ul>		
12. Lernziele:	<p>Studierende entwickeln ein Verständnis für Modelle, Methoden und Konzepte der Mensch-Computer-Interaktion. Sie lernen verschiedene Ansätze für den Entwurf, die Entwicklung und Bewertung von Benutzungsschnittstellen kennen und verstehen deren Vor- und Nachteile.</p>		
13. Inhalt:	<p>Die Vorlesung vermittelt Konzepte, Prinzipien, Modelle, Methoden und Techniken für die effektive Entwicklung von benutzerfreundlichen Mensch-Computer-Schnittstellen. Das Thema moderner Benutzungsschnittstellen wird dabei für klassische Computer aber auch für mobile Geräte, eingebettete Systeme, Automobile und intelligente Umgebungen betrachtet.</p> <p>Die folgenden Themen werden in der Vorlesung behandelt:</p> <ul style="list-style-type: none"> <li>• Einführung in die Grundlagen der Mensch-Computer Interaktion, historische Entwicklung</li> <li>• Entwurfsprinzipien und Modelle für moderne Benutzungsschnittstellen und interaktive Systeme</li> <li>• Informationsverarbeitung des Menschen, Wahrnehmung, Motorik, Eigenschaften und Fähigkeiten des Benutzers</li> <li>• Interaktionskonzepte und -stile, Metaphern, Normen, Regeln und Style Guides</li> </ul>		

- Ein- und Ausgabegeräte, Entwurfsraum für interaktive Systeme
  - Analyse-, Entwurfs- und Entwicklungsmethoden und -werkzeuge für Benutzungsschnittstellen
  - Prototypische Realisierung und Implementierung von interaktiven Systemen, Werkzeuge
  - Architekturen für interaktive Systeme, User Interface Toolkits und Komponenten
  - Akzeptanz, Evaluationsmethoden und Qualitätssicherung
- 

14. Literatur:

- Bernhard Preim, Raimund Dachsel. Interaktive Systeme 1: Grundlagen, Graphical User Interfaces, Informationsvisualisierung. Springer, Berlin; 2. Auflage. 2010
  - Alan Dix, Janet Finley, Gregory Abowd, Russell Beale, Human-Computer Interaction, 2004
  - Ben Shneiderman, Catherine Plaisant, Designing the User Interfaces, 2005
- 

15. Lehrveranstaltungen und -formen:

- 102101 Vorlesung Mensch-Computer-Interaktion
  - 102102 Übung Mensch-Computer-Interaktion
- 

16. Abschätzung Arbeitsaufwand:

Präsenzzeit:	42 h
Selbststudiums- /	138 h
Nachbearbeitungszeit:	
Summe:	180 h

---

17. Prüfungsnummer/n und -name:

- 10211 Mensch-Computer-Interaktion (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein
  - V Vorleistung (USL-V), schriftlich, eventuell mündlich
- 

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von:

Institut für Visualisierung und Interaktive Systeme

---

## Modul: 10220 Modellierung

2. Modulkürzel:	052010001	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Bernhard Mitschang</li> <li>• Frank Leymann</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog SWT →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog SWT →</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• 051520005 Programmierung und Software-Entwicklung</li> <li>• 051510005 Datenstrukturen und Algorithmen</li> <li>• 051200005 Systemkonzepte und -programmierung</li> </ul>		
12. Lernziele:	<p>Am Ende des Moduls sind die Studierenden in der Lage, wesentliche Artefakte eines IT Systems zu modellieren. Der Zusammenhang und das Zusammenspiel solcher Artefakte ist verstanden. Die Rolle von Metamodellen und deren Erstellung ist klar.</p>		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Entity-Relationship Modell &amp; komplexe Objekte</li> <li>• Relationenmodell &amp; Relationenalgebra , Überblick SQL</li> <li>• Transformationen von ER nach Relationen, Normalisierung</li> <li>• XML, DTD, XML-Schema, Info-Set, Namensräume</li> <li>• Metamodelle &amp; Repository</li> <li>• RDF, RDF-S &amp; Ontologien</li> <li>• UML</li> <li>• Petri Netze, Workflownetze</li> <li>• BPMN</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• A. Silberschatz, H. F. Korth, S. Sudarshan, Database System Concepts, 2002</li> <li>• R. Eckstein, S. Eckstein, "XML und Datenmodellierung", dpunkt.verlag 2004</li> <li>• M. Hitz, G. Kappel, E. Kapsammer, W. Retschitzegger, UML @ Work - Objektorientierte Modellierung mit UML2, 2005</li> <li>• P. Hitzler, M. Krötzsch, S. Rudolph, Y. Sure, Semantic Web, 2008</li> <li>• T.J. Teorey, Database Modeling &amp; Design, 2nd Edition, 1994</li> </ul>		

- H.J. Habermann, F. Leymann, "Repository", Oldenbourg 1993
- W. Reisig, "Petri-Netze", Vieweg & Teubner 2010
- B. Silver, "BPMN Method & Style", Cody-Cassidy Press 2009

---

15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"><li>• 102201 Vorlesung Modellierung</li><li>• 102202 Übung Modellierung</li></ul>
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"><li>• 10221 Modellierung (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein</li><li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich</li></ul>
18. Grundlage für ... :	<ul style="list-style-type: none"><li>• 10030 Architektur von Anwendungssystemen</li><li>• 10080 Datenbanken und Informationssysteme</li></ul>
19. Medienform:	
20. Angeboten von:	Institut für Architektur von Anwendungssystemen

---

## Modul: 39040 Rechnernetze

2. Modulkürzel:	051200010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:		Univ.-Prof. Kurt Rothermel	
9. Dozenten:		<ul style="list-style-type: none"> <li>• Kurt Rothermel</li> <li>• Frank Dürr</li> </ul>	
10. Zuordnung zum Curriculum in diesem Studiengang:		<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p>	
11. Empfohlene Voraussetzungen:		<ul style="list-style-type: none"> <li>• 051520005 Programmierung und Software-Entwicklung</li> <li>• 051510005 Datenstrukturen und Algorithmen</li> <li>• Grundkenntnisse in Java</li> </ul>	
12. Lernziele:		<ul style="list-style-type: none"> <li>• Versteht grundsätzliche Eigenschaften, Konzepte und Methoden von Rechnernetzen, insbesondere dem Internet.</li> <li>• Versteht Schichten und deren Zusammenwirken in einem Protokollstapel</li> <li>• Kann Rechnernetze aufbauen, verwalten und analysieren.</li> <li>• Kann Protokolle entwickeln und in Schichtenarchitektur einbetten.</li> <li>• Kann höhere Kommunikationsdienste zur Entwicklung von netzgestützten Systemen anwenden.</li> <li>• Kann sich mit Experten anderer Domänen über Methoden der Rechnernetze verständigen.</li> </ul>	
13. Inhalt:		<ul style="list-style-type: none"> <li>• Einführung in die Rechnernetze, ISO Referenzmodell;</li> <li>• Bitübertragungsschicht: Übertragungsmedien, analoge und digitale Informationskodierung und -übertragung, Vermittlungsarten;</li> <li>• Sicherungsschicht: Betriebsarten, Fehlererkennung und -behandlung, Flusskontrolle;</li> <li>• Lokale Netze: CSMA/CD, Token Ring, Token Bus, FDDI, Kopplung;</li> <li>• Vermittlungsschicht: Verbindungsorientierter und verbindungsloser Dienst, Leitwegbestimmung, Überlastkontrolle;</li> <li>• Internetworking;</li> <li>• Internet-Protokoll;</li> <li>• Transportschicht: ausgewählte Realisierungsprobleme und Internet-Protokolle;</li> <li>• Echtzeitkommunikation: IntServ, DiffServ; Sicherheit: Verfahren, IPsec, SSL, TLS.</li> </ul>	
14. Literatur:		<ul style="list-style-type: none"> <li>• A.S. Tanenbaum, Computer Networks, 4th Edition, 2003</li> <li>• D.E. Comer, Computernetzwerke und Internets, 2000</li> <li>• D.E. Comer, Internetworking with TCP/IP Volume I: Principles, Protocols, and Architecture, 1995</li> </ul>	

- J. F. Kurose, K. W. Ross, Computer Networks: a top-down approach featuring the Internet, 2001
  - L.L. Peterson, B.S. Davie, Computer Networks: A Systems Approach, 1999
- 

15. Lehrveranstaltungen und -formen:

- 390401 VL Rechnernetze
  - 390402 ÜB Rechnernetze
- 

16. Abschätzung Arbeitsaufwand:

Präsenzzeit: 42 Stunden  
Nachbearbeitungszeit: 138 Stunden

---

17. Prüfungsnummer/n und -name:

- 39041 Rechnernetze (PL), schriftlich oder mündlich, 90 Min.,  
Gewichtung: 1.0, Prüfungsdauer: 90 min schriftlich oder 30 min mündlich Exam duration: 90 min written exam or 30 min oral exam
  - V Vorleistung (USL-V), schriftlich, eventuell mündlich
- 

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von:

Verteilte Systeme

---

## Modul: 36530 Rechnerorganisation 1

2. Modulkürzel:	051700006	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Hans-Joachim Wunderlich		
9. Dozenten:	Hans-Joachim Wunderlich		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog SWT →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog SWT →</p>		
11. Empfohlene Voraussetzungen:	Einführung in die Technische Informatik (14360)		
12. Lernziele:	<ul style="list-style-type: none"> <li>• Kenntnisse über die Grundlagen der Organisation von Rechnern und deren maschinennahe Programmierung</li> <li>• Grundzüge über die Beschreibung und den Entwurf von Hardwaresystemen</li> </ul>		
13. Inhalt:	<p>Aufbau, Entwurf und maschinennahe Programmierung von Mikroprozessoren werden erläutert. In den Übungen wird das Wissen durch schriftliche Aufgaben sowie Experimente mit Prozessorsimulatoren vertieft. Im einzelnen werden behandelt:</p> <ul style="list-style-type: none"> <li>• Grundstrukturen: Stack-, Akkumulator- und Register-basierende Rechner</li> <li>• Informationsdarstellung in Hardware: Daten- und Befehlsformate, Fehlererkennung und -korrektur, Befehlssätze und Unterstützung von Hochsprachen.</li> <li>• MIPS als RISC-Bespiel und seine Assemblerprogrammierung</li> <li>• Grundelemente und Entwurf auf Register-Transfer-Ebene</li> <li>• Grundzüge einer Hardware-Beschreibungssprache</li> <li>• Operationswerke: Multiplikation, Division, Gleitkommaeinheiten</li> <li>• Steuerwerksentwurf und Mikroprogrammierung</li> <li>• Befehlszyklus und Unterbrechungen</li> <li>• Pipelining und statisches Scheduling</li> <li>• Speicherorganisation: Cachestrukturen und virtueller Speicher, Seitenverwaltung, Segmentierung, TLB, MMU und DMA</li> <li>• Leistungsbewertung: Maßzahlen und CPI, Benchmarking und einfache Anwendung von Warteschlangen</li> </ul>		

14. Literatur:	Literatur, siehe Webseite zur Veranstaltung								
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"><li>• 365301 Vorlesung Rechnerorganisation 1</li><li>• 365302 Praktische und theoretische Übungen Rechnerorganisation 1</li></ul>								
16. Abschätzung Arbeitsaufwand:	<table><tr><td>Präsenzzeit:</td><td>42 h</td></tr><tr><td>Selbststudiums- /</td><td>138 h</td></tr><tr><td>Nachbearbeitungszeit:</td><td></td></tr><tr><td>Summe:</td><td>180 h</td></tr></table>	Präsenzzeit:	42 h	Selbststudiums- /	138 h	Nachbearbeitungszeit:		Summe:	180 h
Präsenzzeit:	42 h								
Selbststudiums- /	138 h								
Nachbearbeitungszeit:									
Summe:	180 h								
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"><li>• 36531 Rechnerorganisation 1 (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0</li><li>• V Vorleistung (USL-V), schriftliche Prüfung</li></ul>								
18. Grundlage für ... :									
19. Medienform:									
20. Angeboten von:	Rechnerarchitektur								

---

## Modul: 40090 Systemkonzepte und -programmierung

2. Modulkürzel:	051200005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Kurt Rothermel		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Kurt Rothermel</li> <li>• Frank Dürr</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule --&gt;Katalog SWT →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog ISG →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog ISW →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --&gt;Katalog SWT →</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>* Modul 051520005 Programmierung und Software-Entwicklung</li> <li>* Modul 051510005 Datenstrukturen und Algorithmen</li> </ul>		
12. Lernziele:	<ul style="list-style-type: none"> <li>* Verstehen grundlegender Architekturen und Organisationsformen von Software-Systemen</li> <li>* Verstehen systemnaher Konzepte und Mechanismen</li> <li>* Kann existierende Systemplattformen und Betriebssysteme hinsichtlich ihrer Eigenschaften analysieren und anwenden.</li> <li>* Kann systemnahe Software entwerfen und implementieren.</li> <li>* Kann nebenläufige Programme entwickeln</li> <li>* Kann mit Experten anderer Fachgebiete die Anwendung von Systemfunktionen abstimmen.</li> </ul>		
13. Inhalt:	<p>Grundlegende Systemstrukturen - und organisationen</p> <ul style="list-style-type: none"> <li>• Multitaskingsystem</li> <li>• Multiprozessorsystem</li> <li>• Verteiltes System</li> </ul> <p>Modellierung und Analyse nebenläufiger Programme</p> <ul style="list-style-type: none"> <li>• Abstraktionen: Atomare Befehle, Prozesse, nebenläufiges Programm</li> <li>• Korrektheit- und Leitungskriterien</li> </ul> <p>Betriebssystemkonzepte</p> <ul style="list-style-type: none"> <li>• Organisation von Betriebssystemen</li> <li>• Prozesse und Threads</li> <li>• Eingabe/Ausgabe</li> <li>• Scheduling</li> </ul>		

Konzepte zur Synchronisation über gemeinsamen Speicher

- Synchronisationsprobleme und -lösungen
- Synchronisationswerkzeuge: Semaphor, Monitor

Konzepte zur Kommunikation und Synchronisation mittels Nachrichtentransfer

- Taxonomie: Kommunikation und Synchronisation
- Nachrichten als Kommunikationskonzept
- Höhere Kommunikationskonzepte

Basialgorithmen für Verteilte Systeme

- Erkennung globaler Eigenschaften
- Schnappschussproblem
- Konsistenter globaler Zustand
- Verteilte Terminierung

Praktische nebenläufige Programmierung in Java

- Threads und Synchronisation
- Socketschnittstelle
- RMI Programmierung

---

14. Literatur:	Literatur, siehe Webseite zur Veranstaltung
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"><li>• 400901 Vorlesung Systemkonzepte und -programmierung</li><li>• 400902 Übung Systemkonzepte und -programmierung</li></ul>
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"><li>• 40091 Systemkonzepte und -programmierung (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0</li><li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich</li></ul>
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Verteilte Systeme

---

## 340 Wahlmodule aus Master SWT

---

Zugeordnete Module:	10040	Bildsynthese
	10080	Datenbanken und Informationssysteme
	10120	Modellbildung und Simulation
	10250	Parallele Systeme
	14380	Hardware Verification and Quality Assessment
	24900	Fachpraktikum Rechnergestützte Methoden der Produktentwicklung
	29430	Computer Vision
	29440	Geometric Modeling and Computer Animation
	29450	Graphentheorie
	29460	Kryptographische Verfahren
	29470	Machine Learning
	29480	Loose Coupling and Message Based Applications
	29500	Visual Computing
	29510	Service Computing
	29550	Algorithmische Geometrie
	29570	Computer Interface Technologien
	29580	Data Compression
	29590	Digitale Systeme
	29600	Digital System Design II
	29610	Hardware Based Fault Tolerance
	29620	Fortgeschrittene Aspekte der Sprachperzeption und Sprachproduktion
	29640	Mikrocontroller
	29650	Parallele Programmierung
	29660	Programmanalysen und Compilerbau
	29670	Rapid Prototyping
	29680	Real-Time Programming
	29690	Real-Time Video Processing I
	29700	Real-Time Video Processing II
	29710	Embedded Systems Engineering
	29720	Mobile Computing
	29730	Modelling, Simulation, and Specification
	29740	Fachpraktikum Eingebettete Systeme
	29750	Fachpraktikum Rechnerarchitektur
	29760	Algorithmische Gruppentheorie
	39250	Distributed Systems I
	40680	Optimization
	42420	High Performance Computing
	42460	Numerische Simulation
	42480	Ausgewählte Kapitel des Wissenschaftlichen Rechnens
	42520	Services and Service Composition
	42840	Software-Recht
	42860	Hauptseminar (Master SWT 1)
	42900	Business Process Management
	42910	Advanced Business Process Management
	42920	Hardware-Software-Codesign
	45730	Distributed Systems II
	45740	Rechnernetze II
	45750	Fachpraktikum Verteilte Systeme
	45760	Ausgewählte Kapitel der Algorithmentheorie
	46660	Service Management and Cloud Computing, and Evaluation
	48480	Data Engineering
	48500	Image Synthesis
	48550	Practical Course Information Systems
	48570	Practical Course Visual Computing

48580 Reinforcement Learning  
48600 Robotics I  
48620 Scientific Visualization  
51720 IT-Strategy  
51740 Quantencomputing  
55560 Hauptseminar (Master SWT 2)  
55600 Advanced Information Management  
55610 Information Integration  
55620 Data Warehousing, Data Mining, and OLAP  
55630 Information Visualization and Visual Analytics  
55640 Correspondence Problems in Computer Vision  
55650 Multimodal Interaction for Ubiquitous Computers  
55740 Advanced Service Computing  
56550 Software Verification  
56680 Automaten über unendlichen Objekten  
56790 Parallele Numerik  
56980 Fachpraktikum Cloud Architekturen und Management  
57050 Compilerbau  
58190 Entwurf und Implementierung eines Compilers  
58440 Fachpraktikum: Algorithmik  
60120 Interaktive Systeme  
60140 Sprachbau mit Language Workbenches  
60860 3D Scanner - Algorithms and Systems

---

## Modul: 60860 3D Scanner - Algorithms and Systems

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:		Univ.-Prof. Sven Simon	
9. Dozenten:			
10. Zuordnung zum Curriculum in diesem Studiengang:		B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module	
11. Empfohlene Voraussetzungen:			
12. Lernziele:			
13. Inhalt:			
14. Literatur:			
15. Lehrveranstaltungen und -formen:	608601	Vorlesung mit Übung 3D-Scanner - Algorithmen und Systeme	
16. Abschätzung Arbeitsaufwand:			
17. Prüfungsnummer/n und -name:	60861	3D Scanner - Algorithms and Systems (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0	
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

## Modul: 42910 Advanced Business Process Management

2. Modulkürzel:	052010007	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Frank Leymann</li> <li>• Dimka Karastoyanova</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	052010006 - Business Process Management		
12. Lernziele:	<p>Am Ende der Veranstaltungen haben die Teilnehmer weiterführende Ansätze zur Modellierung von Prozessen und zur Spezifikation von Workflows verstanden. Die Rolle von Muster in der Beschreibung von Workflows ist klar geworden. Verfahren des Process Mining sind theoretisch dargestellt. Die Notwendigkeit zur P2P-Verzahnung ("Choreographien") von Prozessen und entsprechende Ansätze sind klar. Ebenso verstanden ist das darüber hinausgehende Konzept der Komponentenverdrahtung. Weitere Architekturen und Einsatzgebiete von WFMS sind verstanden.</p>		
13. Inhalt:	<p>In der Vorlesung werden fortgeschrittene Themen des Workflowmanagement vorgestellt. Aktuelle Entwicklungen aus dem Forschungsumfeld und der Industrie auf dem Gebiet werden diskutiert.</p> <p>Human Task Management          Weitere Ansätze zur Prozessmodellierung (Pi-Kalkül, WSFL, XLANG,...)          Muster (Kontrol-, Datenfluss, Organisatorisch)          Process Monitoring          Process Mining          Peer-to-Peer Verzahnung von Prozessen (Choreographie, Gebrauchsanweisungen,...)          Verdrahten von Komponenten (Global Models,...)          Anwendungsbereiche (Manufakturing, Compliance,...)          Prozessadaption und -flexibilität</p>		
14. Literatur:	W. van der Aalst, K. van Hee, Workflow Management, 2002		
15. Lehrveranstaltungen und -formen:	429101 Vorlesung mit Übungen, Workflow Management 2		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 42911 Advanced Business Process Management (PL), schriftlich oder mündlich, 60 Min., Gewichtung: 1.0</li> <li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich</li> </ul>		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Architektur von Anwendungssystemen		

## Modul: 55600 Advanced Information Management

2. Modulkürzel:	051200099	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	PD Holger Schwarz		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Holger Schwarz</li> <li>• Bernhard Mitschang</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	Lecture "Modellierung" or comparable course		
12. Lernziele:	The students learn current concepts for modeling, developing, deploying and processing database-oriented applications. This includes technologies and standards for XML processing and their integration into database systems as well as concepts and systems for content management and data management in the cloud.		
13. Inhalt:	<p>Among the topics to be discussed in this course are:</p> <ul style="list-style-type: none"> <li>• XML and database technology (XML modeling, XML storage, XML query languages, XML processing)</li> <li>• NoSQL data management (Key value stores, MapReduce, triple stores, document stores, graph stores)</li> <li>• Content management (Enterprise content management, information retrieval, search technologies)</li> </ul>		
14. Literatur:	Will be announced at the beginning of the lecture.		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 556001 Vorlesung Advanced Information Management</li> <li>• 556002 Übung Advanced Information Management</li> </ul>		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit: 42 Stunden          Selbststudium: 138 Stunden</p> <p><b>Gesamt: 180 Stunden</b></p>		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 55601 Advanced Information Management (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0, schriftlich (90 min) oder mündlich (20 min), written (90 min) or oral (20 min)</li> <li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich, 90 Min.</li> </ul>		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Datenbanken und Informationssysteme		

## Modul: 55740 Advanced Service Computing

2. Modulkürzel:	052010005	5. Moduldauer:	2 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	5.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Dimka Karastoyanova		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Dimka Karastoyanova</li> <li>• Frank Leymann</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	<p>Service Computing, Lecture and Exercise (4 SWS)          or          Services and Service Composition, Lecture and Exercise (4SWS)</p>		
12. Lernziele:	<p>This module comprises two lectures and therefore topics from two areas of advanced service computing. The focus of the Lecture Advanced Service Computing is concepts and technologies for describing and providing stateful resources as Web Services as well as the use of Semantics in Web Services and service compositions. The focus in the Lecture Services and Security is on security aspects of service-based applications.</p>		
13. Inhalt:	<p>This module comprises two lectures and therefore topics from two areas of advanced service computing.</p> <p>Based on the topics discussed in the lecture Service Computing, in the Lecture Advanced Service Computing we will focus on concepts and technologies for describing and providing stateful resources as Web Services. In this respect we will also consider Grid Services and infrastructures. In addition, the topics Semantic Web, Ontologies and Semantic Web Services will be presented in detail. Particular attention will be paid to Semantic Web Service Technologies and frameworks like OWL-S, WSMO, SAWSDL and approaches for their use in service compositions.</p> <p>The focus in the Lecture Services and Security is on security aspects of service-based applications. Foundations of Security in enterprise architectures will be presented, as well as best practices for enterprise and IT security in terms of patterns. Basic Security approaches (e.g. prevention, detection, reaction) and mechanisms (access control, authentication, identification, cryptography) will be presented in detail. We will also discuss current state of the art of Web application and Web Service security.</p>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Literatur, die begleitende Literatur wird in der Veranstaltung und im Web bekannt gegeben.</li> <li>• S. Graham, D. Davis, S. Simeonov, G. Daniels, P. Brittenham, Y. Nakamura, P. Fremantle, D. König, C., Building Web Services with Java (2nd Edition), 2005</li> <li>• S. Weerawarana, F. Curbera, F. Leymann, T. Storey, D. Ferguson, Web Services Platform Architecture, 2005</li> <li>- Markus Schumacher et al.: Security Patterns: Integrating Security and Systems Engineering, Wiley Series in Software Design Patterns, 2004</li> </ul>		



## Modul: 29550 Algorithmische Geometrie

2. Modulkürzel:	050410105	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Stefan Funke		
9. Dozenten:	Stefan Funke		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Grundkenntnisse in Datenstrukturen und Algorithmen wie sie in "Datenstrukturen und Algorithmen" (Modul 12060), "Algorithmen und Berechenbarkeit" (Modul 11890), und "Algorithmik" (Modul 10020) vermittelt werden.		
12. Lernziele:	Die Teilnehmer kennen die Grundbegriffe der Algorithmischen Geometrie und haben einen Überblick über die Methoden und Techniken, die in der Algorithmischen Geometrie angewandt werden.		
13. Inhalt:	Es werden die grundlegenden Techniken und Methoden der Algorithmischen Geometrie vermittelt.		
14. Literatur:	<ul style="list-style-type: none"> <li>de Berg, M., Cheong, O., van Kreveld, M., Overmars, M.: Computational Geometry - Algorithms and Applications, Springer, 2000.</li> </ul>		
15. Lehrveranstaltungen und -formen:	295501 Vorlesung Algorithmische Geometrie		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	29551 Algorithmische Geometrie (PL), schriftlich oder mündlich, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Algorithmik		

## Modul: 29760 Algorithmische Gruppentheorie

2. Modulkürzel:	050420115	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Apl. Prof. Ulrich Hertrampf		
9. Dozenten:	Volker Diekert		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Elementare Gruppentheorie		
12. Lernziele:	Die Studierenden beherrschen typische Denk- und Herangehensweisen aus der algorithmischen und kombinatorischen Gruppentheorie. Sie wissen, wie man diverse algorithmische Probleme in freien Gruppen mit Hilfe der Stallingsgraphen lösen kann. Sie können mit Darstellungen von Gruppen durch Erzeugende und Relationen umgehen. Sie kennen das Wortproblem und deren Lösung für gewisse Klassen von Gruppen. Sie kennen konfluente Ersetzungssysteme, HNN-Erweiterungen, amalgamierte Produkte und die Grundbegriffe der Bass-Serre-Theorie.		
13. Inhalt:	Bereits 1911 formulierte Max Dehn drei fundamentale algorithmische Probleme für endlich dargestellte Gruppen: <ol style="list-style-type: none"> <li>1) Ist ein gegebenes Gruppenelement <math>g</math> (als Wort in Erzeugern) das Einselement in der Gruppe <math>G</math>?</li> <li>2) Sind zwei Elemente <math>g</math> und <math>h</math> konjugiert?</li> <li>3) Definieren zwei gegebene Darstellungen isomorphe Gruppen?</li> </ol> Im Allgemeinen sind alle diese Fragen unentscheidbar, also kann man positive Antworten nur in Spezialfällen erhalten. Bei der Lösung des Wortproblems und bei Strukturaussagen ist vor allem die Technik der konfluenten Wortersetzungssysteme hilfreich, die auch in anderen Bereichen zum Einsatz kommen. Insgesamt lebt die Theorie von Querbezügen zu anderen Bereichen, wie Kombinatorik, Topologie, Geometrie, theoretischer Informatik. Dieses Zusammenspiel verschiedener Methoden macht die algorithmische Gruppentheorie sehr attraktiv.		
14. Literatur:	<ul style="list-style-type: none"> <li>• Björner, Brenti: Combinatorics of Coxeter groups, Springer, 2005.</li> <li>• Camps, Große Rebel, Rosenberger: Einführung in die kombinatorische und geometrische Gruppentheorie, Heidemannm Verlag 2008.</li> <li>• Lyndon, Schupp: Combinatorial Group Theory, Springer, 1977.</li> <li>• Magnus, Karrass, Solitar: Combinatorial Group Theory, Wiley &amp; Sons, 1966.</li> <li>• Serre: Trees, Springer, 1980.</li> <li>• Stillwell: Classical Topology and Combinatorial Group Theory, Springer, 1993.</li> </ul>		
15. Lehrveranstaltungen und -formen:	297601 Vorlesung mit Übung Algorithmische Gruppentheorie		

16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h
	Selbststudiums- /	138 h
	Nachbearbeitungszeit:	
	Summe:	180 h
<hr/>		
17. Prüfungsnummer/n und -name:	29761 Algorithmische Gruppentheorie (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0	
<hr/>		
18. Grundlage für ... :		
<hr/>		
19. Medienform:		
<hr/>		
20. Angeboten von:	Theoretische Informatik	
<hr/>		

## Modul: 45760 Ausgewählte Kapitel der Algorithmtheorie

2. Modulkürzel:	050410115	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Apl. Prof. Ulrich Hertrampf		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Volker Diekert</li> <li>• Stefan Funke</li> <li>• Ulrich Hertrampf</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Grundkenntnisse in Algorithmen und Komplexität		
12. Lernziele:	Die Teilnehmer lernen aktuellste Resultate aus der Algorithmtheorie kennen.		
13. Inhalt:	Es werden aktuelle Forschungsergebnisse in der Algorithmtheorie präsentiert.		
14. Literatur:	aktuelle wissenschaftliche Originalartikel		
15. Lehrveranstaltungen und -formen:	457601 Vorlesung Ausgewählte Kapitel der Algorithmtheorie		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	45761 Ausgewählte Kapitel der Algorithmtheorie (PL), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Formale Methoden der Informatik		

## Modul: 42480 Ausgewählte Kapitel des Wissenschaftlichen Rechnens

2. Modulkürzel:	051240030	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Dirk Pflüger		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Dirk Pflüger</li> <li>• Stefan Zimmer</li> <li>• Miriam Mehl</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 10190 Mathematik für Informatiker und Softwaretechniker <b>und</b></li> <li>• Modul 10240 Numerische und Stochastische Grundlagen der Informatik <b>bzw</b> .</li> <li>• Modul 41590 Einführung in die Numerik und Stochastik für Softwaretechniker</li> <li>• Modul 42410 Grundlagen des wissenschaftlichen Rechnens</li> </ul>		
12. Lernziele:	Die Teilnehmer kennen ausgewählte aktuelle Forschungsthemen des wissenschaftlichen Rechnens und können mit der zugehörigen Primärliteratur arbeiten.		
13. Inhalt:	Aktuelle weiterführende Forschungsthemen des wissenschaftlichen Rechnens, wie z.B. adaptive Finite Elemente, hierarchische Basen und dünne Gitter, robuste Multilevellöser, Wavelets und schnelle Wavelettransformation, p-Version oder Spektralverfahren.		
14. Literatur:	<p>Primärliteratur zu den behandelten Themen:</p> <ul style="list-style-type: none"> <li>• Bungartz/Griebel: Sparse Grids; Acta Numerica, Volume 13, p. 147-269</li> <li>• Quarteroni/Valli: Numerical approximation of partial differential equations</li> <li>• Quarteroni: Numerical models for differential problems</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 424801 Vorlesung Ausgewählte Kapitel des Wissenschaftlichen Rechnens</li> <li>• 424802 Übung Ausgewählte Kapitel des Wissenschaftlichen Rechnens</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	42481 Ausgewählte Kapitel des Wissenschaftlichen Rechnens (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Simulation großer Systeme		

## Modul: 56680 Automaten über unendlichen Objekten

2. Modulkürzel:	050420230	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Apl. Prof. Ulrich Hertrampf		
9. Dozenten:	Volker Diekert		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	Grundkenntnisse in Mathematik und Theoretischer Informatik. (reguläre Sprachen und endliche Automaten).		
12. Lernziele:	<p>Die Studierenden lernen die wichtigsten Grundtechniken in dem Bereich der formalen Verifikation für nicht terminierende Systeme und nebenläufige Prozess kennen. Sie lernen Denkweisen und Resultate aus verschiedenen mathematischen Disziplinen wie der Topologie, der Logik, oder der Kombinatorik kennen. Sie kennen den Begriff der MSO-Logik und ihre Entscheidbarkeit nach Büchi und Rabin.</p>		
13. Inhalt:	<p>Die Vorlesung behandelt eine mathematischen Theorie für nicht terminierende Systeme und nebenläufige Prozess. Bei der formalen Verifikation kommen Automatenmodelle zum Einsatz, welche unendliche Objekte als Eingabe erhalten. So lassen sich viele Methoden von endlichen Wörtern auf weitere Bereiche wie unendliche Sequenzen oder Bäume ausdehnen. In diesem Sinne ist die Automatentheorie über unendlichen Objekten wesentlich reichhaltiger und spannender als über endlichen Wörtern. Die Vorlesung orientiert sich an den folgenden Themen:</p> <ul style="list-style-type: none"> <li>• Presburger Arithmetik: Anforderungen an Automaten</li> <li>• Büchi Automaten und omega-reguläre Sprachen</li> <li>• Klarlunds Konstruktion zur Komplementierung von Büchi Automaten</li> <li>• Andere Akzeptanzbedingungen für omega-Automaten</li> <li>• Monadische Logik zweiter Stufe (MSO)</li> <li>• Deterministische omega-Sprachen</li> <li>• Topologisch definierte Sprachklassen</li> <li>• McNaughtons Theorem</li> <li>• Die Safra-Konstruktion</li> <li>• Algebraische Beschreibungen</li> <li>• Eindeutige Büchi Automaten</li> <li>• Logik erster Stufe und andere Fragmente von MSO</li> <li>• Paritätsspiele</li> <li>• Automaten über unendlichen Bäumen</li> <li>• Rabins Baumtheorem</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Volker Diekert, Manfred Kufleitner, Gerhard Rosenberger: Diskrete algebraische Methoden: Arithmetik, Kryptographie, Automaten und Gruppen. De Gruyter, Berlin 2013.</li> <li>• Volker Diekert und Paul Gastin: First-order definable languages. In Jörg Flum, Erich Grädel, Thomas Wilke (eds.). Logic and Automata:</li> </ul>		

History and Perspectives. Texts in Logic and Games 2, Amsterdam University Press 2008, pp. 261-306.

- Wolfgang Thomas: Automata on infinite objects. In Jan van Leeuwen (ed.). Handbook of Theoretical Computer Science, volume B: Formal Models and Semantics. Elsevier, 1990, pp. 133-192.
- Wolfgang Thomas: Languages, Automata, and Logic. In Grzegorz Rozenberg and Arto Salomaa (eds). Handbook of Formal Languages, volume 3: Beyond Words. Springer, New York, 1997, pp. 389-455.

---

15. Lehrveranstaltungen und -formen:	566801 Vorlesung Automaten über unendlichen Objekten	
<hr/>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h
	Selbststudiums- /	138 h
	Nachbearbeitungszeit:	
	Summe:	180 h
<hr/>		
17. Prüfungsnummer/n und -name:	56681 Automaten über unendlichen Objekten (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0	
<hr/>		
18. Grundlage für ... :		
<hr/>		
19. Medienform:		
<hr/>		
20. Angeboten von:	Institut für Formale Methoden der Informatik	

---

## Modul: 10040 Bildsynthese

2. Modulkürzel:	051900012	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Thomas Ertl		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Martin Fuchs</li> <li>• Thomas Ertl</li> <li>• Daniel Weiskopf</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 10060 Computergraphik</li> </ul>		
12. Lernziele:	<p>Die Studierenden besitzen Wissen über verschiedene Ansätze und Algorithmen der dreidimensionalen Computergraphik, physikalisch-basierte Verfahren wie Raytracing und Radiosity, die den Lichttransport und die Wechselwirkung mit Materie modellieren, und numerische Methoden wie Monte-Carlo-Integration und Finite-Elemente-Verfahren die es erlauben, die Rendering-Gleichung zu lösen. Darüber hinaus kennen sie interaktive Verfahren, die unter Ausnutzung programmierbarer Grafik-Hardware realistische Beleuchtungseffekte in Echtzeit approximieren können, sowie bildbasierte Ansätze, die ohne geometrische Daten realistische Darstellungen erzeugen. Bild-basierte Verfahren verzichten auf eine geometrische Repräsentation der Szene und erzeugen neue Ansichten aus anderen aufgenommenen Bildern.</p>		
13. Inhalt:	<p>In dieser Vorlesung werden die folgenden Themen behandelt:</p> <ul style="list-style-type: none"> <li>• Grafik Hardware und APIs, OpenGL</li> <li>• Texturen, prozedurale Modelle</li> <li>• Schattenberechnungen</li> <li>• Szenengraphen, Culling, Level-of-Detail Verfahren</li> <li>• Physikalisch-basierte Beleuchtungsberechnung, Fotorealistische Bildsynthese</li> <li>• Lokale Beleuchtungsmodelle</li> <li>• Raytracing, Monte-Carlo Methoden</li> <li>• Radiosity</li> <li>• Bild-basiertes Rendering</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Andrew S. Glassner: Principles of Digital Image Synthesis, 1995</li> <li>• D. Eberly: 3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics, 2000</li> <li>• J. Foley, A. van Dam, S. Feiner, J. Hughes: Computer Graphics: Principle and Practice, 1990</li> <li>• Literatur, siehe Webseite zur Veranstaltung</li> <li>• P. Dutre, P. Bekaert, K. Bala: Advanced Global Illumination, 2003</li> <li>• Tomas Akenine-Möller, Eric Haines: Real-Time Rendering, 2002</li> <li>• Matt Pharr, Greg Humphreys: Physically Based Rendering: From Theory To Implementation, Morgan Kaufmann; Auflage: 2nd revised edition. (26. August 2010)</li> </ul>		

- Peter Shirley et al: Fundamentals of Computer Graphics, Third Edition, A.K. Peters, July 2009
- 

15. Lehrveranstaltungen und -formen:

- 100401 Vorlesung Bildsynthese
- 100402 Übung Bildsynthese

---

16. Abschätzung Arbeitsaufwand:

Präsenzzeit:	42 h
Selbststudiums- /	138 h
Nachbearbeitungszeit:	
Summe:	180 h

---

17. Prüfungsnummer/n und -name:

- 10041 Bildsynthese (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0
- V Vorleistung (USL-V), schriftlich, eventuell mündlich, Übungsschein.

---

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von: Institut für Visualisierung und Interaktive Systeme

---

## Modul: 42900 Business Process Management

2. Modulkürzel:	052010006	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Frank Leymann</li> <li>• Dimka Karastoyanova</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 5. Semester            → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT            →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester            → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	611 Grundlagen der Architektur von Anwendungssystemen, Vorlesung mit Übung, 4,0 SWS		
12. Lernziele:	<p>The course has the objective to provide knowledge about the essential modelling constructs for workflows and their mapping to corresponding workflow languages. In addition, the life cycle of Workflow-based applications will be presented in detail and connected to the Architecture of Workflow Management Systems, which will also be presented. Moreover, the goal is to enable students to use workflow languages (in particular BPEL) in practice. In this respects students will also understand the fundamental approach process graphs, which is applied in workflow languages. Of great importance are , mechanisms for fault handling and exception handling - these will be explained in detail and students will be able to apply them.</p>		
13. Inhalt:	<p>Workflows are IT realisations of business processes and are also considered an approach of significant importance for composition of applications. This course will introduce the foundations of this area, also known as Business Process Management BPM).</p> <ol style="list-style-type: none"> <li>1. Historical Development of the Workflow Technology</li> <li>2. Business Re-engineering (BPM Lifecycle, Tools,...)</li> <li>3. Architecture of WFMS (Navigator, Executor, Worklist Manager,...)</li> <li>4. Flow Languages (FDL, BPEL)</li> <li>5. Process Model Graph (mathematical meta-model: syntax, operational semantics)</li> <li>6. Advanced funcitons (sub-processes, event handling, instance modifications, adaptation)</li> <li>7. Two-level programming paradigm</li> <li>8. Transactional support in workflows</li> </ol>		
14. Literatur:	<ul style="list-style-type: none"> <li>• F. Leymann, D. Roller, Production Workflow, 2000</li> <li>• W. van der Aalst, K. van Hee, Workflow Management, 2002</li> </ul>		
15. Lehrveranstaltungen und -formen:	429001 Vorlesung mit Übungen, Workflow Management 1		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit: 42 Stunden            Selbststudium: 138 Stunden</p>		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 42901 Business Process Management (PL), schriftlich oder mündlich, 60 Min., Gewichtung: 1.0, Wenn Prüfung in diesem Fach, dann K E I N E Prüfung in Services and Service</li> </ul>		

- V Composition (auch nicht innerhalb der VTL) möglich! mündl.  
Prüfungsdauer: 30 Minuten  
Vorleistung (USL-V), schriftlich, eventuell mündlich
- 

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von: Institut für Architektur von Anwendungssystemen

---

## Modul: 57050 Compilerbau

2. Modulkürzel:	051010201	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Erhard Plödereder		
9. Dozenten:	Erhard Plödereder		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	<p>Grundkenntnisse aus den Einführungsvorlesungen des Informatikgrundstudiums, sowie einige Erfahrungen mit Programmierung. Vorkenntnisse über formale Sprachen sind vorteilhaft, aber nicht zwingend.</p>		
12. Lernziele:	<p>Die Studierenden haben die Grundkenntnisse erlangt, die zur effizienten Verwendung von Lexer- und Parser-Generatoren zur Analyse von Eingabetexten nötig sind. Sie verstehen die grundlegende Funktionsweise mehrerer Parse-Verfahren und kennen deren grammatikalischen Einschränkungen. Sie kennen elementare Verfahren semantischer Analysen und sind in der Lage, einfache semantische Prüfungen zu verfassen. Sie haben gelernt, die Fehlermeldungen aus Parser-Generatoren, Compilern oder Interpretern richtig einzuordnen. Ferner haben sie durch Betrachtung der Implementierungsmodelle typischer Programmiersprachenkonstrukte Verständnis für das Ausführungsverhalten erlangt. Sie kennen elementare Begriffe der Codegenerierung und die Eigenschaften von typischen Zwischencodedarstellungen in Compilern.</p>		
13. Inhalt:	<p>Compilerarchitekturen im Überblick; lexikalische und syntaktische Analyse von Texten mit formaler Grammatik, insb. von Programmiersprachen. Lexikalische Analyse: endliche Automaten und ihre Implementierung; Syntexanalyse: diverse Parser- Strategien, ihre Implementierung und Eigenschaften. Methoden der automatischen Generierung von Analysatoren aus Spezifikationen der Grammatiken. Fehlererkennung und -behandlung. Analyse der statischen Semantik: Grundbegriffe und elementare Methoden. Attributgrammatiken. Zwischencodeerzeugung. Realisierung einiger Aspekte der Laufzeitsemantik prozeduraler Programmiersprachen. Einfache Codegenerierung.</p>		
14. Literatur:	<ul style="list-style-type: none"> <li>• A. V. Aho, M.S. Lam, R. Sethi, J. D. Ullman: Compilers - Principles, Techniques, and Tools, Addison Wesley Verlag (2007)</li> <li>• Niklaus Wirth: Compilerbau: Eine Einführung, Teubner Verlag (1986)</li> <li>• Wilhelm, Maurer: Übersetzerbau, 2. Auflage, Springer Verlag (1997)</li> <li>• Andrew W. Appel: Modern Compiler Implementation In Java, Cambridge University Press (2002)</li> <li>• Uwe Kastens: Übersetzerbau, Oldenbourg Verlag (1990)</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 570501 Vorlesung Compilerbau</li> <li>• 570502 Übung Compilerbau</li> </ul>		

16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h
	Selbststudiums- /	138 h
	Nachbearbeitungszeit:	
	Summe:	180 h
17. Prüfungsnummer/n und -name:	57051 Compilerbau (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0	
18. Grundlage für ... :	29660 Programmanalysen und Compilerbau	
19. Medienform:		
20. Angeboten von:	Programmiersprachen und ihre Übersetzer	

---

## Modul: 29570 Computer Interface Technologien

2. Modulkürzel:	051230105	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Kenntnisse in mindestens einem Fach der Technischen Informatik oder ähnlichen Fächern und Erfahrungen in mindestens einer Programmiersprache.		
12. Lernziele:	Die Studierenden haben die Funktionsweise und den Aufbau von Coputer Interfaces verstanden. Sie beherrschen verschiedene Interface-Konzepte und kennen die Eigenschaften der Datenströme wie Latenzzeit, tatsächliche Durchsatzrate, Echtzeitfähigkeit, Umgang mit Übertragungsfehlern, etc.		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Grundlagen - Computer Interfaces</li> <li>• Computer Interfaces und OSI-Modelle</li> <li>• Bus- und Netz-Topologien</li> <li>• Line und Error Codes</li> <li>• Protokolle</li> <li>• Treiber</li> <li>• Compliance Tests</li> <li>• Standardization Groups: USB, PCI, etc.</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Patterson, David A.; Hennessey, John L., Computer Organization and Design - The Hardware / Software Interface, 2008</li> </ul> <p>More literature is named in the lecture.</p>		
15. Lehrveranstaltungen und -formen:	295701 Vorlesung mit Übung Computer Interface Technologien		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden  <b>Gesamt: 180 Stunden</b>		
17. Prüfungsnummer/n und -name:	29571 Computer Interface Technologien (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0, Schriftliche Prüfung von 120 Minuten oder mündliche Prüfung von 30 Min.		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

## Modul: 29430 Computer Vision

2. Modulkürzel:	051900215	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Andrés Bruhn		
9. Dozenten:	Andrés Bruhn		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 10190 Mathematik für Informatiker und Softwaretechniker</li> <li>• Modul 10170 Imaging Science</li> </ul>		
12. Lernziele:	<p>Der Student / die Studentin beherrscht die Grundlagen der Merkmalsextraktion und -repräsentation, des 3-D Maschinensehens, der Bildsegmentierung sowie der Mustererkennung. Er/sie kann Probleme aus dem Fachgebiet einordnen und diese selbständig mit den erlernten Algorithmen und Verfahren lösen.</p> <p>The student knows the basics of feature extraction and representation, 3-D computer vision, image segmentation and pattern recognition. He/ she can solve problems of the field using the methods discussed in the course.</p>		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Lineare Diffusion, Skalenräume</li> <li>• Bildpyramiden, Kanten und Eckendetektion</li> <li>• Hough-Transformation, Invarianten</li> <li>• Texturanalyse</li> <li>• Scale Invariant Feature Transform (SIFT)</li> <li>• Bildfolgenanalyse: lokale Verfahren</li> <li>• Bewegungsmodelle, Objektverfolgung, Feature Matching</li> <li>• Bildfolgenanalyse: globale Verfahren</li> <li>• Kamerageometrie, Epipolargeometrie</li> <li>• Stereo Matching und 3-D Rekonstruktion</li> <li>• Shape-from-Shading</li> <li>• Isotrope und anisotrope nichtlineare Diffusion</li> <li>• Segmentierung mit globalen Verfahren</li> <li>• Kontinuierliche Morphologie, Schockfilter</li> <li>• Mean Curvature Motion</li> <li>• Self-Snakes, Aktive Konturen</li> <li>• Bayes'sche Entscheidungstheorie der Mustererkennung</li> <li>• Klassifikation mit parametrischen Verfahren, Dichteschätzung</li> <li>• Klassifikation mit nicht-parametrischen Verfahren</li> <li>• Dimensionsreduktion</li> </ul> <ul style="list-style-type: none"> <li>• Linear Diffusion, Scale Space</li> <li>• Image Pyramids, Edges and Corners</li> <li>• Hough Transform, Invariants</li> <li>• Texture Analysis</li> <li>• Scale Invariant Feature Transform</li> <li>• Image Sequence Analysis: Local Methods</li> <li>• Motion Models, Tracking, Feature Matching</li> </ul>		

- Image Sequence Analysis: Variational Methods
- Camera Geometry, Epipolar Geometry
- Stereo Matching and 3-D Reconstruction
- Shape-from-Shading
- Isotropic and Anisotropic Nonlinear Diffusion
- Segmentation with Global Methods
- Continuous Scaled Morphology, Shock Filters
- Mean Curvature Motion
- Self-Snakes, Active Contours
- Bayes Decision Theory for Pattern Recognition
- Classification with Parametric Techniques, Density Estimation
- Classification with Non-Parametric Techniques
- Dimensionality Reduction

---

14. Literatur:	<ul style="list-style-type: none"> <li>• Forsyth, David and Ponce, Jean: Computer Vision. A Modern Approach, 2003</li> <li>• Bigun, J.: Vision with Direction, 2006</li> <li>• L. G. Shapiro, G. C. Stockman: Computer Vision, 2001</li> <li>• O. Faugeras, Q.-T. Luong: The Geometry of Multiple Images, 2001.</li> </ul>								
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 294301 Vorlesung Computer Vision</li> <li>• 294302 Übung Computer Vision</li> </ul>								
16. Abschätzung Arbeitsaufwand:	<table border="0" style="width: 100%;"> <tr> <td style="width: 60%;">Präsenzzeit:</td> <td style="text-align: right;">42 h</td> </tr> <tr> <td>Selbststudiums- /</td> <td style="text-align: right;">138 h</td> </tr> <tr> <td>Nachbearbeitungszeit:</td> <td></td> </tr> <tr> <td>Summe:</td> <td style="text-align: right;">180 h</td> </tr> </table>	Präsenzzeit:	42 h	Selbststudiums- /	138 h	Nachbearbeitungszeit:		Summe:	180 h
Präsenzzeit:	42 h								
Selbststudiums- /	138 h								
Nachbearbeitungszeit:									
Summe:	180 h								
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 29431 Computer Vision (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben</li> <li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich</li> </ul>								
18. Grundlage für ... :	55640 Correspondence Problems in Computer Vision								
19. Medienform:									
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme								

---

## Modul: 55640 Correspondence Problems in Computer Vision

2. Modulkürzel:	051900211	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	6.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Andrés Bruhn		
9. Dozenten:	Andrés Bruhn		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 10190 Mathematik für Informatiker und Softwaretechniker</li> <li>• Modul 10170 Imaging Science</li> <li>• Modul 29430 Computer Vision</li> </ul>		
12. Lernziele:	<p>Der Student kann Korrespondenzprobleme im Computer-Vision-Bereich selbständig einordnen, Lösungsstrategien mathematisch modellieren und diese dann geeignet algorithmisch umsetzen.</p> <p>The student has knowledge on the different correspondence problems in computer vision, is able to develop mathematical models for solution strategies and implement the corresponding algorithms in an appropriate way.</p>		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Basisverfahren: Block Matching, Detektion von Verdeckungen, Merkmalsfindung, Feature Matching</li> <li>• Optischer Fluss: Lokale und Globale differentiale Verfahren, Parametrisierungsmodelle, Konstanzannahmem, Daten- und Glattheitsterme, Numerik, Große Verschiebungen, Hochgenaue Verfahren</li> <li>• Stereorekonstruktion: Projektive Geometrie, Epipolargeometrie, Schätzung der Fundementalmatrix</li> <li>• Szenenfluss: Gemeinsame Schätzung von Struktur, Bewegung und Geometrie</li> <li>• Medizinische Bildregistrierung: Mutual Information, Elastische und krümmungsbasierte Regularisierung, Landmarks</li> <li>• Particle Image Velocimetry: Div-Curl-Regularisierung, Inkompressibler Navier Stokes Prior</li> </ul> <ul style="list-style-type: none"> <li>• Basic Approaches: Block Matching, Occlusion Detection, Interest Points, Feature Matching</li> <li>• Optic Flow: Local and Global Differential Methods, Parametrisation Models, Constancy Assumptions, Data and Smootness Terms, Numerics, Large Displacements, High Accuracy Methods</li> <li>• Sterep Matching: Projective Geometry, Epipolar Geometry, Estimation of the Fundamental Matrix</li> <li>• Scene Flow: Joint Estimation of Structure, Motion, and Geometry</li> <li>• Medical Image Registration: Mutual Informaion, Elastic and Curvature-Based Regularisation, Landmarks</li> <li>• Particle Image Velocimetry: Div-Curl-Regularisation, Incompressible Navier Stokes Prior</li> </ul>		

14. Literatur:	<ul style="list-style-type: none"><li>• O. Faugeras, Q.-T. Luong: The Geometry of Multiple Images, 2001.</li><li>• J. Modersitzki: Numerical Methods for Image Registration, 2003.</li><li>• A. Bruhn: Variational Optic Flow Computation: Accurate Modeling and Efficient Numerics, Ph.D. Thesis, 2006.</li></ul>								
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"><li>• 556401 Vorlesung Correspondence Problems in Computer Vision</li><li>• 556402 Übung Correspondence Problems in Computer Vision</li></ul>								
16. Abschätzung Arbeitsaufwand:	<table><tr><td>Präsenzzeit:</td><td>42 h</td></tr><tr><td>Selbststudiums- /</td><td>138 h</td></tr><tr><td>Nachbearbeitungszeit:</td><td></td></tr><tr><td>Summe:</td><td>180 h</td></tr></table>	Präsenzzeit:	42 h	Selbststudiums- /	138 h	Nachbearbeitungszeit:		Summe:	180 h
Präsenzzeit:	42 h								
Selbststudiums- /	138 h								
Nachbearbeitungszeit:									
Summe:	180 h								
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"><li>• 55641 Correspondence Problems in Computer Vision (PL), schriftlich, eventuell mündlich, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben</li><li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich</li></ul>								
18. Grundlage für ... :									
19. Medienform:									
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme								

---

## Modul: 29580 Data Compression

2. Modulkürzel:	051230110	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	This course requires basic knowledge in mathematics.		
12. Lernziele:	The students learn the concepts of data compression and acquire an understanding of different algorithms for data compression. Furthermore they will be able to implement and further develop the algorithms discussed in the course.		
13. Inhalt:	Topic of the lecture are algorithms and hardware architectures for data compression <ul style="list-style-type: none"> <li>• Shannon Entropy</li> <li>• Huffman coding</li> <li>• Universal codes</li> <li>• Arithmetic coding</li> <li>• Lossy and Lossless compression</li> <li>• Image data compression</li> <li>• Dictionary based compression</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Khalid Sayood, Introduction to Data Compression, 2005</li> <li>• More literature is named in the lecture</li> </ul>		
15. Lehrveranstaltungen und -formen:	295801 Vorlesung mit Übung Datenkompression		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden <b>Gesamt: 180 Stunden</b>		
17. Prüfungsnummer/n und -name:	29581 Data Compression (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, written 90 Min. or oral 30 Min.		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

## Modul: 48480 Data Engineering

2. Modulkürzel:	051210011	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Melanie Herschel		
9. Dozenten:	Melanie Herschel		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 3. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 3. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Lecture "Modellierung" or comparable course		
12. Lernziele:	The students learn the basic concepts of modeling and system-related issues in data engineering in general and with respect to specific application areas in research-related and engineering-related areas. The methodological basis is defined by information extraction and information analysis, all based on effective metadata management.		
13. Inhalt:	Data engineering involves any data processing necessary to prepare data for subsequent use, e.g., for data analytics or learning. This lecture covers foundations, algorithms, and systems on selected topics of data engineering. These include: <ul style="list-style-type: none"> <li>• Data collection: how do we find relevant data sources?</li> <li>• Big Data integration: Given the unique properties of big data, how can data from multiple data sources be combined to get a more global perspective on a subject to be analyzed?</li> <li>• Data cleaning:How can important properties and errors of data be assessed and corrected?</li> <li>• Data distribution:What modern technologies support the wide dissemination of data?</li> <li>• Provenance: How can the whole data engineering process be documented, controlled, and improved leveraging so-called provenance meta-data?</li> </ul>		
14. Literatur:	There is no unique book covering all aspects of data engineering. The lecture is however significantly based on selected chapters of the following books. <ul style="list-style-type: none"> <li>• Xin Luna Dong and Divesh Srivastava.Big Data Integration. Synthesis Lectures on Data Management, Morgan an Claypool, 2015.</li> <li>• Wanfei Fan and Floris Geerts.Fondations of Data Quality Management. Synthesis Lectures on Data Management, Morgan an Claypool, 2012.</li> <li>• AnHai Doan, Alon Halevy, and Zachary Ives. Principles of Data Integration. Morgan Kaufmann, 2012.</li> <li>• James Cheney, Laura Chiticariu, and Wang Chiew Tan.Provenance in Databases: Why, How, and Where. Foundations and Trends in Databases, Vol. 1, No.4, 2007.</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 484801 Lecture Data Engineering</li> <li>• 484802 Exersice Data Engineering</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden		

Selbststudium: 138 Stunden

---

17. Prüfungsnummer/n und -name: 48481 Data Engineering (PL), schriftlich oder mündlich, 90 Min.,  
Gewichtung: 1.0

---

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von: Data Engineering

---

## Modul: 55620 Data Warehousing, Data Mining, and OLAP

2. Modulkürzel:	051210105	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Bernhard Mitschang		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Bernhard Mitschang</li> <li>• Holger Schwarz</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	Lecture "Modellierung" or comparable course		
12. Lernziele:	<p>After attending this lecture, students understand the challenges behind the integration of heterogeneous data sources in consolidated warehouses and the provisioning of analytical services. They know the typical data warehouse architecture as well as current trends, e.g., real-time data warehousing. Further topics are the structure of a data warehouse and the main processes for building data warehouses (extraction, transformation, load). A special focus is on technologies to analyze data warehouse data, e.g., reporting, online analytic processing and data mining, and their role as part of analytical services.</p>		
13. Inhalt:	<p>Among the topics to be discussed in this course are:</p> <ul style="list-style-type: none"> <li>- Introduction to data warehousing</li> <li>- Data warehouse architecture</li> <li>- Data warehouse design</li> <li>- Extraction, transformation, load</li> <li>- ETL as a service</li> <li>- Introduction to analytics and analytic services</li> <li>- Real-time reporting</li> <li>- Online analytic processing</li> <li>- Data mining</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• A. Kemper, A. Eickler, Datenbanksysteme - Eine Einführung, 2004</li> <li>• H. Garcia-Molina, J. D. Ullman, J. Widom, Database Systems. The Complete Book, 2003</li> </ul> <p>Will be announced at the beginning of the lecture</p>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 556201 Vorlesung Data Warehousing, Data Mining und OLAP-Technologien</li> <li>• 556202 Übung Data Warehousing, Data Mining und OLAP-Technologien</li> </ul>		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit: 42 Stunden          Selbststudium: 138 Stunden</p> <p><b>Gesamt: 180 Stunden</b></p>		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 55621 Data Warehousing, Data Mining, and OLAP (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0,</li> </ul>		

- V Vorleistung (USL-V), schriftlich oder mündlich, Übungsleistungen während der Unterrichtsperiode als Voraussetzung für die Zulassung zur Prüfung.
- 

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von: Datenbanken und Informationssysteme

---

## Modul: 10080 Datenbanken und Informationssysteme

2. Modulkürzel:	051200025	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Bernhard Mitschang		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Bernhard Mitschang</li> <li>• Holger Schwarz</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	Vorlesung Modellierung oder Gleichwertiges		
12. Lernziele:	Die Studierenden haben die erforderlichen Kenntnisse für Datenbankprogrammierer in angemessenem Umfang erworben.		
13. Inhalt:	<p>Die Vorlesung "Datenbanken und Informationssysteme" ist als Einstiegsveranstaltung in das Vertiefungsgebiet Datenbanksysteme konzipiert. Aufbauend auf dem Inhalt der Vorlesung "Modellierung" werden insbesondere Entwurfs- und Realisierungsaspekte von Datenbanksystemen betrachtet. Die Entwicklung, Installation und Administration von Datenbanksystemen bestimmen hier sowohl Stoffauswahl als auch Detaillierungsgrad.</p> <p>Als Grundlage für alle weiteren Betrachtungen wird ein Schichtenmodell zur Beschreibung eines allgemeinen Datenbanksystems vorgestellt. Darauf aufbauend werden die einzelnen Systemschichten im Detail diskutiert, die dort zu realisierenden Komponenten betrachtet sowie die jeweils vorherrschenden Algorithmen beschrieben und bewertet. Im Einzelnen werden folgende Aspekte vertieft: Anwendungsprogrammierschnittstelle, Externspeicherverwaltung, DBS-Pufferverwaltung, Speicherungsstrukturen und Zugriffspfadstrukturen, Anfrageverarbeitung und Anfrageoptimierung, Transaktionsverarbeitung, Synchronisation, Logging und Recovery.</p>		
14. Literatur:	<ul style="list-style-type: none"> <li>• A. Kemper, A. Eickler, Datenbanksysteme - Eine Einführung, 2004</li> <li>• Th. Härder, E. Rahm, Datenbanksysteme, 2008</li> <li>• H. Garcia-Molina, J. D. Ullman, J. Widom, Database Systems. The Complete Book, 2003</li> <li>• R. Elmasri, S. Navathe, Fundamentals of Database Systems, 2003</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 100801 Vorlesung Datenbanken und Informationssysteme</li> <li>• 100802 Übung Datenbanken und Informationssysteme</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 Stunden	
	Selbststudium:	138 Stunden	

17. Prüfungsnummer/n und -name: 10081 Datenbanken und Informationssysteme (PL), schriftlich oder mündlich, 60 Min., Gewichtung: 1.0, Prüfungsvorleistung: Modalitäten werden in der ersten Vorlesung angegeben

---

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von:

---

## Modul: 29600 Digital System Design II

2. Modulkürzel:	051230122	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	This lectures requires the knowledge of "System Design I". Alternatively, knowledge of "Technische Informatik" is sufficient to follow the course.		
12. Lernziele:	The students will learn to build and implement a complex digital system by using digitals components on a circuit board, and will acquire an in-depth knowledge for implementing complex digital systems using FPGA's.		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Presentation of a case study of a digital system</li> <li>• Simulatable specification of the system</li> <li>• Architecture for Implementation using FPGAs</li> <li>• Design and design tools for board integration</li> <li>• Implementation of a digital system</li> <li>• Verification of a digital system</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Kou-Chuan Chang, K. C. Chang, Digital Systems Design with VHDL and Synthesis: An Integrated Approach, 1999</li> <li>• More literature is named in the lecture</li> </ul>		
15. Lehrveranstaltungen und -formen:	296001 Vorlesung mit Übung Digital System Design II		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	29601 Digital System Design II (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

## Modul: 29590 Digitale Systeme

2. Modulkürzel:	051230120	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Kenntnisse in einem Fach aus der Technischen Informatik oder einem ähnlichen Gebiet.		
12. Lernziele:	Die Studierende beherrschen den Entwurf Digitaler Systeme durch die Integration von digitalen Komponenten auf einem Board und die Realisierung von digitaler Komponenten mittels FPGAs.		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Praktische Einführung in den System-Entwurf mit digitalen Komponenten wie Schnittstellenbausteinen zur Kommunikation, FPGAs, Prozessoren, intelligenten Sensoren etc.</li> <li>• Einführung und Verwendung der Hardware-Beschreibungssprache VHDL zum Entwurf Digitaler Systeme</li> <li>• Digitale Systeme und Board-Integration von digitalen Komponenten</li> <li>• Aufbau von Computer-Boards u. Gbit/s-Interconnects</li> <li>• Entwurf auf höheren Abstraktionsebenen zur schnellen Entwicklung von Prototypen</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Kou-Chuan Chang, K.C. Chang, Digital Systems Design with VHDL and Synthesis: An Integrated Approach, 1999</li> </ul> <p>More literature is named in the lecture.</p>		
15. Lehrveranstaltungen und -formen:	295901 Vorlesung mit Übung Digital System Design I		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden  <b>Gesamt: 180 Stunden</b>		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 29591 Digitale Systeme (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0, Schriftliche Prüfung von 120 Min. oder mündliche Prüfung von 30 Min.</li> <li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich</li> </ul>		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

## Modul: 39250 Distributed Systems I

2. Modulkürzel:	051200015	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Kurt Rothermel		
9. Dozenten:	Kurt Rothermel		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Programmierung und Software-Entwicklung Datenstrukturen und Algorithmen Systemkonzepte und -Programmierung		
12. Lernziele:	The Students will gain an understanding of the basic characteristics, concepts and methods of distributed systems. Furthermore, the ability to analyze existing distributed applications and platforms with regard to its specific properties will be obtained. The implementation of distributed applications as well as system platforms based on the shown methods of that course is another objective. Due to the knowledge provided in that course, the students will be able to communicate with other experts of other professional disciplines, about topics in the field of distributed systems.		
13. Inhalt:	1. Introduction to distributed systems 2. System models 3. Communication: Messages, Remote Procedure Call (RPC), Remote Method Invocation RMI 4. Naming: Generating and Resolution 5. Time Management and clocks in distributed Systems: Applications, logical clocks, physical clocks, synchronization of clocks 6. Global state: concepts, snapshot algorithms, distributed Debugging 7. Transaction management: Serializability, barrier methods, 2-phase-commit-protocols 8. Data replication: primary copy, consensus-protocols and other algorithms 9. Safety/Security: Methods for confidentiality, integrity, authentication and authorization 10. Multicast-algorithms: processing model, broadcast-semantics and algorithms		
14. Literatur:	Literatur, siehe Webseite zur Veranstaltung		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 392501 Vorlesung Verteilte Systeme</li> <li>• 392502 Übungen Verteilte Systeme</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 h Selbststudiumszeit / Nachbearbeitungszeit: 138 h Gesamt: 180 h		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 39251 Distributed Systems I (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0, Prüfungsdauer: 90 min schriftlich oder 30</li> </ul>		

- V min mündlich Exam duration: 90 min written exam or 30 min oral exam  
Vorleistung (USL-V), schriftlich, eventuell mündlich
- 

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von: Verteilte Systeme

---

## Modul: 45730 Distributed Systems II

2. Modulkürzel:	051200169	5. Moduldauer:	2 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Kurt Rothermel		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Kurt Rothermel</li> <li>• Muhammad Tariq</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	The Lecture requires basic knowledge from the course Distributed Systems I		
12. Lernziele:	<p>In this lecture, the aquired knowledge from the previous lecture Distributed Systems I is depend. The student will gain information about further practice-oriented problems and will implement protocols to solve those problems. The student will be capable to analyze distributed systems in terms of these problems, design, apply and develop protocols for specific applications.</p>		
13. Inhalt:	<ol style="list-style-type: none"> <li>1. Group communication</li> <li>2. Consensus</li> <li>3. Fault tolerant services</li> <li>4. Wave algorithms</li> <li>5. Termination</li> <li>6. Garbage collection</li> <li>7. Election</li> <li>8. Deadlocks</li> <li>9. Organisational &amp; Introduction</li> </ol>		
14. Literatur:	<ul style="list-style-type: none"> <li>• J.L. Welch, H. Attiya, Distributed Computing: Fundamentals, Simulations and Advanced Topics, 1997</li> </ul> <p>The event is based on a collection of scientific papers, which will be announced in the lecture.</p>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 457301 Vorlesung Verteilte Algorithmen</li> <li>• 457302 Vorlesung Asynchronous Middleware Systems</li> </ul>		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit: 42 Stunden          Selbststudium: 138 Stunden</p> <p><b>Gesamt: 180 Stunden</b></p>		
17. Prüfungsnummer/n und -name:	45731 Distributed Systems II (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0, Prüfungsdauer: 90 min schriftlich oder 30 min mündlichExam duration: 90 min written exam or 30 min oral exam		
18. Grundlage für ... :			
19. Medienform:			

20. Angeboten von:

Verteilte Systeme

---

## Modul: 29710 Embedded Systems Engineering

2. Modulkürzel:	051711027	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Martin Radetzki		
9. Dozenten:	Martin Radetzki		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	keine		
12. Lernziele:	Master-level understanding of the design methodology and advanced design techniques for constructing and analyzing embedded hardware / software systems.		
13. Inhalt:	1. Introduction to embedded systems and their design constraints 2. Synthesis models and algorithms 3. System level synthesis 4. High level synthesis 5. Pipelined data path and controller design 6. Software task scheduling and schedulability analysis 7. Static and dynamic methods for scheduling and priority assignment 8. Communication architectures for embedded systems		
14. Literatur:	Skript „Embedded Systems Engineering“ G. Buttazzo: Hard Real Time Computing Systems. 2nd edition, Springer, 2005 P. Eles, K. Kuchcinski, Z. Peng: System Synthesis with VHDL. Kluwer Academic Publishers, 1998. P. Marwedel: Embedded Systems Design. Springer, 2006		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 297101 Vorlesung Embedded Systems Engineering</li> <li>• 297102 Übung Embedded Systems Engineering</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden  <b>Summe: 180 Stunden</b>		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 29711 Embedded Systems Engineering (Klausur) (PL), schriftlich, eventuell mündlich, 120 Min., Gewichtung: 1.0</li> <li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich, Als Zulassungsvoraussetzung zur Klausur ist folgende Vorleistung zu erbringen: Teilnahme an den Übungen, Präsentation der Lösung wenigstens einer Aufgabe.</li> </ul>		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Eingebettete Systeme (Embedded Systems Engineering)		

## Modul: 58190 Entwurf und Implementierung eines Compilers

2. Modulkürzel:	05151313	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Erhard Plödereder		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Erhard Plödereder</li> <li>• Timm Felden</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	<p>Modul Compilerbau ist notwendige Voraussetzung, Java-Kenntnisse werden erwartet.</p> <p>Die Teilnehmerzahl in diesem Modul ist auf maximal 15 beschränkt.</p>		
12. Lernziele:	<p>Die Studierenden haben praktische Erfahrungen mit der Konstruktion eines Compilers und der Umsetzung von Konzepten in Programmiersprachen erworben. Sie sind in der Lage aktuelle Entwicklungen im Bereich der Programmiersprachen und des Compilerbaus zu beurteilen. Durch die Teilnahme an Programmierübungen mit Codereviews haben sie gelernt, qualitativ hochwertige Compiler zu entwickeln.</p>		
13. Inhalt:	<p>Lexer- und Parsergeneratoren, Semantische Attributierung, Fehlererkennung und -behandlung in Compilern, Typsysteme und Typprüfung, Die Java Virtual Machine, Zwischencodegenerierung, Sprachinterfaces</p>		
14. Literatur:	<p>A.W. Appel : Modern Compiler Implementation in Java 2nd Edition; Cambridge University Press (2002)</p> <p>A. V. Aho, M.S. Lahm, R. Sethi, J. D. Ullman: Compilers - Principles, Techniques, and Tools; Addison, Wesley (2007)</p>		
15. Lehrveranstaltungen und -formen:	581901 Vorlesung Entwurf und Implementierung eines Compilers		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit: 42 Stunden</p> <p>Selbststudium: 138 Stunden</p>		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 58191 Entwurf und Implementierung eines Compilers (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0</li> <li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich</li> </ul>		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

## Modul: 56980 Fachpraktikum Cloud Architekturen und Management

2. Modulkürzel:	[pord.modulcode]	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	Johannes Wettinger		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Obligatory: Object-oriented programming (e.g. Java), XML Optional: scripting languages (e.g. Python, Ruby, Bash), JSON, YAML		
12. Lernziele:	The goal of this course is twofold: First, established reference architectures are discussed and used to model scalable architectures for Cloud applications. Second, based on these architectures, different Cloud management approaches are used to provision infrastructure resources, deploy application components, and manage them. The course is targeted at students of MSc Computer Science, MSc Softwaretechnik, and MSc Informatik.		
13. Inhalt:	Architectures for scalable, elastic, and reliable Cloud applications Cloud infrastructure management (OpenStack, Amazon Web Services) Configuration management (Chef, Puppet, CFEngine, etc.) Container virtualization (Docker, LXC, etc.) PaaS-centric management (Cloud Foundry, BOSH, etc.) Model-driven Cloud management: infrastructure-centric models (CloudFormation, Heat, etc.), application-centric models (Juju, TOSCA, etc.)		
14. Literatur:			
15. Lehrveranstaltungen und -formen:	569801 Fachpraktikum Cloud Architekturen und Management		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	56981 Fachpraktikum Cloud Architekturen und Management (LBP), Studienbegleitend, 0 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

## Modul: 29740 Fachpraktikum Eingebettete Systeme

2. Modulkürzel:	051711135	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	6.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Martin Radetzki		
9. Dozenten:	wiss. MA		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Modul "Embedded Systems Engineering"		
12. Lernziele:	Ability to model, simulate, and synthesize digital circuits and systems using state-of-the-art description languages and modelling styles for various abstraction levels.		
13. Inhalt:	1) Introduction to VHDL 2) Dataflow modelling 3) Behavioural modelling 4) Structural modelling 5) VHDL library concept 6) RTL modelling 7) Synthesis 8) Introduction to SystemC 9) Hierarchy and basic communication 10) Channels, events, time, and simulation mechanisms 11) Advanced communication modelling 12) Transaction level modelling (TLM) 13) The OSCI TLM 2.0 standard		
14. Literatur:	Lab handouts Documentation of development tools (provided in the lab)  Peter Ashenden: The Designer's Guide to VHDL (book available in the lab)  Black et al.: SystemC from the Ground Up (book available in the lab)		
15. Lehrveranstaltungen und -formen:	297401 Übung Fachpraktikum Eingebettete Systeme		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit im Labor: 64 Stunden Vor- und Nachbereitung: 116 Stunden  <b>Summe: 180 Stunden</b>		
17. Prüfungsnummer/n und -name:	29741 Fachpraktikum Eingebettete Systeme (LBP), schriftlich oder mündlich, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Eingebettete Systeme (Embedded Systems Engineering)		

## Modul: 29750 Fachpraktikum Rechnerarchitektur

2. Modulkürzel:	051700025	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Hans-Joachim Wunderlich		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Hans-Joachim Wunderlich</li> <li>• Chang Liu</li> <li>• Laura Rodriguez Gomez</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 41930 Rechnerorganisation</li> <li>• Modul 10140 Advanced Processor Architecture</li> </ul>		
12. Lernziele:	Students are able to design digital systems by using the complete state of the art design automation tool chain.		
13. Inhalt:	<p>In this lab course, the students design and implement a RISC processor and extend it with techniques common for high-performance processors. Hardware structures found in the state of the art processors will be applied and adapted.</p> <p>In order to achieve high frequency, proper design and verification techniques play an important role. The students learn how timing analysis, pipelining and retiming can be used to optimize the synthesis results. Because software has to be specifically tailored to such a processor architecture, the lab course also deals with scheduling techniques that allow to avoid pipeline stalls and hazards.</p>		
14. Literatur:	<ul style="list-style-type: none"> <li>• D. A. Patterson, J. L. Hennessy: Computer Organization &amp; Design . The Hardware / Software Interface (3rd Edition); San Francisco, Ca.: Morgan Kaufmann Publishers Inc., 2004</li> <li>• J. L. Hennessy and D. A. Patterson: Computer Architecture - A Quantitative Approach (3rd Edition); San Francisco, Ca.: Morgan Kaufmann Publishers Inc., 2012</li> </ul>		
15. Lehrveranstaltungen und -formen:	297501 Fachpraktikum Rechnerarchitektur		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	29751 Fachpraktikum Rechnerarchitektur (LBP), schriftlich oder mündlich, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Rechnerarchitektur		

## Modul: 24900 Fachpraktikum Rechnergestützte Methoden der Produktentwicklung

2. Modulkürzel:	051400006	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes Semester
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Dieter Roller		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Dieter Roller</li> <li>• Akram Chamakh</li> <li>• Julian Eichhoff</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 4. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	Grundkenntnisse wie sie in „CAD und Produktmodelle“ vermittelt werden.		
12. Lernziele:	<p>Die Studierenden sind nach erfolgreichem Besuch des Moduls in der Lage</p> <ul style="list-style-type: none"> <li>• typische Problemstellungen, Methoden und Technologien im Anwendungsbereich des jeweils behandelten CAx-Feldes sowie dessen Eingliederung in die Produktentwicklungskette unter Verwendung von Fachvokabular zu beschreiben</li> <li>• die vorgestellten Methoden und Technologien gegenüberzustellen und ihren Einsatz im Bezug zu vorgegebenen Problemstellungen zu begründen</li> <li>• die Funktionen eines Technologievertreters aus dem CAx-Feld bei einer vorgegebenen Problemstellung effektiv anwenden können</li> </ul>		
13. Inhalt:	<p>Jedes Semester wechselnd werden zu einem konkreten CAx-Bereich (CAD, CAM, CAP, CAQ) folgende Inhalte behandelt:</p> <ul style="list-style-type: none"> <li>• Eingliederung des Unternehmensbereichs hinsichtlich Produktentstehung/Produktlebenszyklus; Funktionen, Prozesse und Tätigkeiten des Unternehmensbereichs; Informationsflüsse (zugrundeliegende Daten, Objekte und Artefakte) und weitere Verbindung zu anderen Unternehmensbereichen</li> <li>• Überblick zu typischen Problemstellungen und darauf anwendbare Methoden im Aufgabenfeld des Unternehmensbereichs; Methodenvergleich</li> <li>• Überblick der marktdominierenden CAx-Technologien zur Unterstützung des Aufgabenfeldes; Technologievergleich</li> <li>• Anwendungsbezogene Detailvorstellung der Funktionen und Repräsentationsformen eines Vertreters dieser Technologien im Rahmen konkreter Beispielprobleme</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• D. Roller. CAD: Effiziente Anpassungs- und Variantenkonstruktion. Springer, Heidelberg, 1995.</li> <li>• S. Vajna, C. Weber, H. Bley, K. Zeman. CAx für Ingenieure - Eine praxisbezogene Einführung. Springer, Heidelberg, 2009.</li> <li>• G. Pahl, W. Beitz, J. Feldhusen, K.-H. Grote. Konstruktionslehre - Grundlagen erfolgreicher Produktentwicklung Methoden und Anwendung. Springer, Heidelberg, 2007.</li> </ul>		

15. Lehrveranstaltungen und -formen: 249001 Übung Fachpraktikum Rechnergestützte Methoden der Produktentwicklung

---

16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h
	Selbststudiums- /	138 h
	Nachbearbeitungszeit:	
	Summe:	180 h

---

17. Prüfungsnummer/n und -name: 24901 Fachpraktikum Rechnergestützte Methoden der Produktentwicklung (LBP), schriftlich oder mündlich, Gewichtung: 1.0

---

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von: Institut für Rechnergestützte Ingenieursysteme

---

## Modul: 45750 Fachpraktikum Verteilte Systeme

2. Modulkürzel:	051200111	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Kurt Rothermel		
9. Dozenten:	Frank Dürr		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Verteilte Systeme, Rechnernetze II		
12. Lernziele:	Die Teilnehmer besitzen die Fähigkeit, verteilte Anwendungen und Dienste zu entwerfen und zu implementieren. Sie besitzen praktische Kenntnisse in der Netzprogrammierung und der Programmierung von Client/Server-Anwendungen. Sie verfügen über praktische Kenntnisse über Technologien und Werkzeugen zur Implementierung und zum Testen verteilter Systeme.		
13. Inhalt:	<ul style="list-style-type: none"> <li>- Socket-Programmierung</li> <li>- Höherwertige Kommunikationsprotokolle und Webservice- Schnittstellen (HTTP &amp; XML/JSON, RPC, SOAP, REST)</li> <li>- Client/Server-Systeme</li> <li>- Peer-to-Peer- und Ad-hoc-Kommunikation</li> <li>- Entwicklungsumgebungen</li> <li>- Test verteilter Systeme</li> </ul>		
14. Literatur:	- A.S. Tanenbaum: Computer Networks, 4th Edition, 2003		
15. Lehrveranstaltungen und -formen:	457501 Fachpraktikum Verteilte Systeme		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden  <b>Gesamt: 180 Stunden</b>		
17. Prüfungsnummer/n und -name:	45751 Fachpraktikum Verteilte Systeme (PL), Sonstiges, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Verteilte Systeme		

## Modul: 58440 Fachpraktikum: Algorithmik

2. Modulkürzel:	[pord.modulcode]	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes Semester
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Stefan Funke		
9. Dozenten:	Stefan Funke		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:			
12. Lernziele:			
13. Inhalt:			
14. Literatur:	wird in der Veranstaltung bekanntgegeben		
15. Lehrveranstaltungen und -formen:	584401 Fachpraktikum Algorithmik		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	58441 Fachpraktikum: Algorithmik (LBP), schriftlich und mündlich, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Formale Methoden der Informatik		

## Modul: 29620 Fortgeschrittene Aspekte der Sprachperzeption und Sprachproduktion

2. Modulkürzel:	052400010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Antje Schweitzer		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Grzegorz Dogil</li> <li>• Antje Schweitzer</li> <li>• Natalie Lewandowski</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 5. Semester            → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT            →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester            → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	14000 Phonetik und Phonologie		
12. Lernziele:	Die Studierenden haben ein detailliertes Verständnis für Theorien der Sprachproduktion und -perzeption entwickelt. Sie sind in der Lage, aktuelle Forschungsarbeiten in diesen Bereichen zu verstehen und kritisch zu bewerten.		
13. Inhalt:	Es werden aktuelle Konferenz- und Zeitschriftenbeiträge aus den Bereichen Sprachperzeption und Sprachproduktion erarbeitet und diskutiert, unter Berücksichtigung theoretischer und/oder praktischer Aspekte.		
14. Literatur:	<ul style="list-style-type: none"> <li>• R.L. Diehl, A.J. Lotto, L.L. Holt, Speech Perception, Annual Review of Psychology, Annual Reviews, 2004.</li> <li>• W.J.M. Levelt, Speaking: From Intention to Articulation, 1989, MIT Press</li> <li>• Konferenz- und Zeitschriftenbeiträge nach Ankündigung in den Vorlesungen.</li> </ul>		
15. Lehrveranstaltungen und -formen:	296201 Vorlesung Advanced Speech Perception und Advanced Speech Production		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit 42 h Selbststudiumszeit 138 h		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 29621 Fortgeschrittene Aspekte der Sprachperzeption und Sprachproduktion (PL), mündliche Prüfung, 20 Min., Gewichtung: 1.0</li> <li>• V Vorleistung (USL-V), Sonstiges</li> </ul>		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Experimentelle Phonetik		

## Modul: 29440 Geometric Modeling and Computer Animation

2. Modulkürzel:	051900010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Daniel Weiskopf		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Daniel Weiskopf</li> <li>• Thomas Ertl</li> <li>• Guido Reina</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	<p>Basic computer graphics, for example:</p> <ul style="list-style-type: none"> <li>• 10060 Computergraphik</li> </ul>		
12. Lernziele:	<p>Students gain an understanding of the fundamental concepts and techniques of geometric modeling and computer animation. This includes theoretical and mathematical foundations, important algorithms, and implementation aspects as well as practical experience with modeling and animation tools such as Maya.</p>		
13. Inhalt:	<p>This course covers foundations and methods for the modeling of scenes and for computer animation. This includes the representation of curves and surfaces, which are used by modeling and animation software for modeling of objects, description of the dynamics of parameters, or keyframe animation. Physically based animation describes motion via kinematic and dynamics laws of mechanics. Applications thereof include particle systems all the way to character animation and deformation.</p> <p>In particular, the following topics are covered:</p> <ul style="list-style-type: none"> <li>• Description and modeling of curves: differential geometry of curves, polynomial curves in general, interpolation, Bezier curves, B-splines, rational curves, NURBS</li> <li>• Description and modeling of surfaces: differential geometry of surfaces, tensor product surfaces, Bezier patches, NURBS, ruled surfaces, Coons pathes</li> <li>• Subdivision schemes: basic concept, convergence and limit process, sudivision curves, subdivision surfaces</li> <li>• Overview of animation techniques</li> <li>• Keyframe animation, inverse kinematics</li> <li>• Physically based animation of points and rigid bodies: kinematics and dynamics</li> <li>• Particle systems: Reeves, flocking and boids, agent-based simulation</li> <li>• Cloth animation: continuum mechanics, mass-spring model, numerical solvers for ordinary differential equations, explicit and implicit integrators</li> <li>• Collision: efficient collision detection, bounding volume hierachies, hierarchical space partitioning, collision handling, sliding and resting contact</li> </ul>		

- Fluid simulation: wave equation, Navier Stokes, level sets, particle level sets
  - Basics of film production: camera, lighting, production process, storyboard
- 

14. Literatur:

- D. Eberly, 3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics. Morgan Kaufmann, 2000
  - G. Farin: Curves and Surfaces for CAGD: A Practical Guide. Morgan Kaufmann, 2002
  - R. Parent: Computer Animation: Algorithms and Techniques. Morgan Kaufmann, 2002
  - W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling: Numerical Recipes - The Art of Scientific Computing. Cambridge University Press, 1986
- 

15. Lehrveranstaltungen und -formen:

294401 Vorlesung mit Übungen Geometrische Modellierung und Animation

---

16. Abschätzung Arbeitsaufwand:

Präsenzzeit: 42 Stunden, Selbststudium: 138 Stunden

---

17. Prüfungsnummer/n und -name:

- 29441 Geometric Modeling and Computer Animation (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0,
  - V Vorleistung (USL-V), schriftlich, eventuell mündlich, Erfolgreiche Teilnahme an Übungen / exercises passed
- 

18. Grundlage für ... :

---

19. Medienform:

Video projector, blackboard, exercises using PCs

---

20. Angeboten von:

Institut für Visualisierung und Interaktive Systeme

---

## Modul: 29450 Graphentheorie

2. Modulkürzel:	050420105	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Apl. Prof. Ulrich Hertrampf		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Ulrich Hertrampf</li> <li>• Volker Diekert</li> <li>• Manfred Kufleitner</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	Grundvorlesungen in theoretischer Informatik		
12. Lernziele:	Die Studierenden beherrschen typische Denk- und Herangehensweisen aus der Graphentheorie. Die Beziehung zwischen diversen Graphparametern werden verstanden, ebenso wie ihre algorithmische Relevanz. Die Eigenschaften der wichtigsten Graphklassen erschließen sich den Studierenden.		
13. Inhalt:	<p>Die Vorlesung behandelt algorithmische Problem und strukturelle Zusammenhänge bei Graphen. Im Einzelnen werden die folgenden Themen behandelt:</p> <ul style="list-style-type: none"> <li>• Eulergraphen</li> <li>• Cographen</li> <li>• Bipartite Graphen</li> <li>• Planare Graphen, Eulerformel, Satz von Kuratowski</li> <li>• Graphparameter</li> <li>• Perfekte Graphen</li> <li>• Graphenfärbungen und der Satz von Ramsey</li> <li>• Extremale Graphentheorie</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Reinhard Diestel: Graphentheorie. Springer, 2010.</li> <li>• Martin Aigner, Günter M. Ziegler: Das BUCH der Beweise. Springer, 2009.</li> <li>• Jacobus H. van Lint, Richard M. Wilson: A Course in Combinatorics. Cambridge University Press, 2nd edition, 2001.</li> </ul>		
15. Lehrveranstaltungen und -formen:	294501 Vorlesung mit Übungen Graphentheorie		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 29451 Graphentheorie (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0</li> <li>• V Vorleistung (USL-V), schriftlich oder mündlich</li> </ul>		
18. Grundlage für ... :			
19. Medienform:			

20. Angeboten von:

Theoretische Informatik

---

## Modul: 29610 Hardware Based Fault Tolerance

2. Modulkürzel:	051710023	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Hans-Joachim Wunderlich		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Hans-Joachim Wunderlich</li> <li>• Michael Kochte</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 10140 Grundlagen der Rechnerarchitektur / Advanced Processor Architecture</li> <li>• Modul 10310 Rechnerorganisation</li> </ul>		
12. Lernziele:	<ul style="list-style-type: none"> <li>• Knowledge of methods for reliability assessment of circuits and systems</li> <li>• Knowledge of the main techniques for implementing fault tolerance</li> <li>• Knowledge how to design fault tolerant circuits and systems</li> </ul>		
13. Inhalt:	<p>Micro- and Nano-electronic systems can exhibit failures both right after production and during their operation. Systems for which safety and security is of concern have to be designed in a way that the desired function can be delivered even if some components fail or produce erroneous outputs. This lecture presents the most important design techniques that allow to tolerate hardware faults up to a certain degree. The topics of the lecture are as follows:</p> <ul style="list-style-type: none"> <li>• Terminology</li> <li>• Measures of fault tolerance</li> <li>• Techniques for structural and time redundancy</li> <li>• Error detection and diagnosis</li> <li>• Fault masking, repair, reconfiguration</li> <li>• Fault-tolerant distributed systems</li> </ul>		
14. Literatur:	<p>Apart from lecture slides, the following books can be used to deepen on the topics of the lecture:</p> <ul style="list-style-type: none"> <li>• I. Koren and C. M. Krishna: Fault-Tolerant Systems, Morgan-Kaufman (2007)</li> <li>• P. K. Lala: Self-Checking and Fault-Tolerant Digital Design, Morgan-Kaufmann Publishers (2001)</li> <li>• D.K. Pradhan: Fault-Tolerant Computer Design, Prentice Hall (1996)</li> <li>• R.N. Rao and E. Fujiwara: Error Control Coding for Computer Systems, Prentice Hall (1989)</li> <li>• M.L. Bushnell and V.D. Agrawal: Essentials of Electronic Testing, Klumer Academic Publishers (2000)</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 296101 Vorlesung Hardware Based Fault Tolerance</li> <li>• 296102 Übung Hardware Based Fault Tolerance</li> </ul>		

16. Abschätzung Arbeitsaufwand:	Presence Time:	42 h
	Self Study:	138 h
	Sum:	180 h
<hr/>		
17. Prüfungsnummer/n und -name:	29611 Hardware Based Fault Tolerance (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0, Writen exam 90 min or Oral exam 30 min	
<hr/>		
18. Grundlage für ... :		
<hr/>		
19. Medienform:	Laptop presentation	
<hr/>		
20. Angeboten von:	Institut für Technische Informatik	
<hr/>		

## Modul: 14380 Hardware Verification and Quality Assessment

2. Modulkürzel:	051700020	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Hans-Joachim Wunderlich		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Hans-Joachim Wunderlich</li> <li>• Michael Kochte</li> <li>• Laura Rodriguez Gomez</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 4. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 10310 Rechnerorganisation oder</li> <li>• Modul 10140 Grundlagen der Rechnerarchitektur</li> </ul>		
12. Lernziele:	Basic knowledge of methodologies and algorithms of functional and formal verification, diagnosis, test and design for testability of integrated circuits		
13. Inhalt:	<p>Complex integrated circuits and systems are hardly designed fault free at first go. Also during production defects and an imperfect yield have to be expected. The course deals with the basic techniques to find and locate faults and defects in the design and in the manufactured, integrated system. The discussed methods are applied with the help of commercial and academic tools in exercises and labs.</p> <p>The course comprises:</p> <ul style="list-style-type: none"> <li>• Validation: Simulation and emulation in different design levels.</li> <li>• Formal verification: Equivalence checking and model checking.</li> <li>• Test: Fault simulation and test generation.</li> <li>• Debug and diagnosis.</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• G. D. Hachtel, F. Somenzi: Logic Synthesis and Verification Algorithms, 2006</li> <li>• K. L. McMillan: Symbolic Model Checking, 1993</li> <li>• L.-T. Wang, C.-W. Wu, X. Wen: VLSI Test Principles and Architectures - Design for Testability, 2006</li> <li>• M. L. Bushnell, V. D. Agrawal: Essentials of Electronic Testing, 2005</li> <li>• R. Drechsler, B. Becker: Graphenbasierte Funktionsdarstellung, 2000</li> <li>• S. Hassoun, T. Sasao: Logic Synthesis and Verification, 2002</li> <li>• S. Minato: Binary Decision Diagrams and Applications for VLSI CAD, 1996</li> <li>• T. Kropf: Introduction to Formal Hardware Verification, 1999</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 143801 Vorlesung Hardware Verification and Quality Assessment</li> <li>• 143802 Übung Hardware Verification and Quality Assessment</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	

17. Prüfungsnummer/n und -name: 14381 Hardware Verification and Quality Assessment (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0

---

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von: Institut für Technische Informatik

---

## Modul: 42920 Hardware-Software-Codesign

2. Modulkürzel:	051711110	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Martin Radetzki		
9. Dozenten:	Martin Radetzki		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Bachelor-Veranstaltung "Grundlagen der Eingebetteten Systeme" oder gleichwertige Kenntnisse		
12. Lernziele:	Ability to conceptualize systems so that an application-specific, optimized trade-off between hardware and software implementation of system functionality is achieved.		
13. Inhalt:	This module deals with the joint design and optimization of hardware and software for pre-defined applications, covering the following topics: 1. Models for system specification 2. Modelling and simulation with the SystemC library 3. Synthesis of system architectures 4. Resource allocation and operation binding 5. Partitioning of functionality among hardware and software 6. Scheduling and schedulability for parallel multi-core architectures 7. Methods for system optimization 8. Application specific instruction set processors (ASIPs) 9. Network-on-Chip (NoC) interconnect architectures		
14. Literatur:	J. Teich, Digitale Hardware/Software-Systeme, 2. Auflage, 2007		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 429201 Vorlesung Hardware-Software-Codesign</li> <li>• 429202 Übung Hardware-Software-Codesign</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	42921 Hardware-Software-Codesign (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Eingebettete Systeme (Embedded Systems Engineering)		

## Modul: 42860 Hauptseminar (Master SWT 1)

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	3.0 LP	6. Turnus:	jedes Semester
4. SWS:	2.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Stefan Wagner		
9. Dozenten:	Dozenten der Informatik		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Variabel: Je nach dem gewählten Seminarthema können Vorkenntnisse aus weiteren Vorlesungen benötigt werden.		
12. Lernziele:	Die Studierenden können sich mit wissenschaftlicher Originalliteratur auseinandersetzen, deren Kernaussagen rezipieren und sich ein spezielles Thema überwiegend im Selbststudium erarbeiten. Sie sind fähig relevante Daten zu sammeln und zu interpretieren und ihre Erkenntnisse einem Fach- und Laienpublikum verständlich zu präsentieren und auf Fragen aus dem Publikum angemessen und sachgerecht zu reagieren. Sie haben gelernt, sich mit einem wissenschaftlichen Thema über einen längeren Zeitraum hinweg auseinander zu setzen und eigenständig aktuelle Hintergrundinformation zu beschaffen. Sie haben generische Kompetenzen erworben, etwa aktiv an einer wissenschaftlichen Diskussion zu einem vorher bekannten Thema teilzunehmen und durch Fragen an den Vortragenden ihr Verständnis zu erweitern. Sie können eine Diskussion leiten und moderieren und sind befähigt, ihre Ergebnisse den Seminarteilnehmern vorzustellen und mit Hilfe moderner Präsentationstechniken zu visualisieren. Sie sind in der Lage, das von ihnen erarbeitete Thema auch schriftlich darzustellen.		
13. Inhalt:	Variabel: Es werden Seminare zu diversen, häufig aktuellen Themen angeboten.  Welche Seminare zugelassen sind, entscheidet die Studienkommission. Zugelassene Seminare werden typischer Weise durch Aushang bekannt gegeben. Die Seminare sind in Größe und Inhalt so gestaltet, dass die generischen Kompetenzen (Schlüsselqualifikationen) der Studierenden entwickelt werden.		
14. Literatur:	Die begleitende Literatur wird in der Veranstaltung und im Web bekannt gegeben.		
15. Lehrveranstaltungen und -formen:	428601 Hauptseminar		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	21 h	
	Selbststudiums- /	69 h	
	Nachbearbeitungszeit:		
	Summe:	90 h	

17. Prüfungsnummer/n und -name: 42861 Hauptseminar (Master SWT 1) (BSL), schriftlich und mündlich,  
Gewichtung: 1.0

---

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von: Institut für Softwaretechnologie

---

## Modul: 55560 Hauptseminar (Master SWT 2)

2. Modulkürzel:	[pord.modulcode]	5. Moduldauer:	1 Semester
3. Leistungspunkte:	3.0 LP	6. Turnus:	jedes Semester
4. SWS:	2.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Stefan Wagner		
9. Dozenten:	Dozenten der Informatik		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Variabel: Je nach dem gewählten Seminarthema können Vorkenntnisse aus weiteren Vorlesungen benötigt werden.		
12. Lernziele:	Die Studierenden können sich mit wissenschaftlicher Originalliteratur auseinandersetzen, deren Kernaussagen rezipieren und sich ein spezielles Thema überwiegend im Selbststudium erarbeiten. Sie sind fähig relevante Daten zu sammeln und zu interpretieren und ihre Erkenntnisse einem Fach- und Laienpublikum verständlich zu präsentieren und auf Fragen aus dem Publikum angemessen und sachgerecht zu reagieren. Sie haben gelernt, sich mit einem wissenschaftlichen Thema über einen längeren Zeitraum hinweg auseinander zu setzen und eigenständig aktuelle Hintergrundinformation zu beschaffen. Sie haben generische Kompetenzen erworben, etwa aktiv an einer wissenschaftlichen Diskussion zu einem vorher bekannten Thema teilzunehmen und durch Fragen an den Vortragenden ihr Verständnis zu erweitern. Sie können eine Diskussion leiten und moderieren und sind befähigt, ihre Ergebnisse den Seminarteilnehmern vorzustellen und mit Hilfe moderner Präsentationstechniken zu visualisieren. Sie sind in der Lage, das von ihnen erarbeitete Thema auch schriftlich darzustellen.		
13. Inhalt:	Variabel: Es werden Seminare zu diversen, häufig aktuellen Themen angeboten.  Welche Seminare zugelassen sind, entscheidet die Studienkommission. Zugelassene Seminare werden typischer Weise durch Aushang bekannt gegeben. Die Seminare sind in Größe und Inhalt so gestaltet, dass die generischen Kompetenzen (Schlüsselqualifikationen) der Studierenden entwickelt werden.		
14. Literatur:	Die begleitende Literatur wird in der Veranstaltung und im Web bekannt gegeben.		
15. Lehrveranstaltungen und -formen:	555601 Hauptseminar		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	21 h	
	Selbststudiums- /	69 h	
	Nachbearbeitungszeit:		
	Summe:	90 h	
17. Prüfungsnummer/n und -name:	55561 Hauptseminar (Master SWT 2) (BSL), schriftlich und mündlich, Gewichtung: 1.0		
18. Grundlage für ... :			

---

19. Medienform:

---

20. Angeboten von: Institut für Softwaretechnologie

---

## Modul: 42420 High Performance Computing

2. Modulkürzel:	051240040	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Dirk Pflüger		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Martin Bernreuther</li> <li>• Dirk Pflüger</li> <li>• Miriam Mehl</li> <li>• Stefan Zimmer</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT →</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 10190 Mathematik für Informatiker und Softwaretechniker <b>und</b></li> <li>• Modul 10240 Numerische und Stochastische Grundlagen der Informatik <b>bzw</b> .</li> <li>• Modul 41590 Einführung in die Numerik und Stochastik für Softwaretechniker</li> </ul>		
12. Lernziele:	<ul style="list-style-type: none"> <li>• Fähigkeit, parallele Algorithmen auf unterschiedlichen parallelen Plattformen mit Hilfe geeigneter algorithmischer Modelle zu bewerten.</li> <li>• Kenntnis verschiedener Programmiermodelle für Parallelrechner mit verteiltem und gemeinsamem Speicher.</li> <li>• Fähigkeit, auch fortgeschrittene Implementierungsaufgaben aus dem Bereich des Höchstleistungsrechnens auf Basis ausgewählter Programmiermodelle zu bewältigen.</li> </ul>		
13. Inhalt:	<p>Die Vorlesung beschäftigt sich mit den Grundlagen paralleler Programmierung und paralleler Algorithmen speziell im Hinblick auf die Anwendungsbereiche Wissenschaftliches Rechnen und High Performance Computing.</p> <p>Verwandte Fragestellungen aus dem Bereich der Theorie (parallele Modelle und parallele Komplexität, etc.) sowie aus der Rechnertechnik (parallele Architekturen) werden begleitend diskutiert.</p> <p>Nach einer allgemeinen Einführung (Klassifizierung von Parallelrechnern, Ebenen von Parallelität, Performance und Architekturen, etc.), werden die Grundlagen paralleler Programme eingeführt (Notation/Syntax, Synchronisation und Kommunikation, Design paralleler Programme, etc.). Sowohl die Programmierung auf Systemen mit gemeinsamem Speicher als auch auf Systemen mit verteiltem Speicher werden besprochen. Dabei wird jeweils mindestens ein geeignetes Programmiermodell (z.B. OpenMP, MPI, CUDA) vertieft behandelt.</p> <p>Aus dem Bereich des High Performance Computing werden begleitend klassische Algorithmen und Implementierungstechniken als Beispiele behandelt, z.B. parallele Algorithmen aus der linearen Algebra (Matrixmultiplikation, etc. oder einfache Verfahren zur Lösung partieller Differentialgleichungen). Zusätzlich können Themen wie Lastverteilung und Lastbalancierung (Grundlagen, Algorithmen zur Partitionierung und Lastbalancierung, etc.) vorgestellt werden.</p>		

14. Literatur:	<ul style="list-style-type: none"><li>• T. Rauber, G. Runger: „Parallele Programmierung“, 2. Aufl., Springer 2007; (in English: T. Rauber, G. Runger: „Parallel Programming: for Multicore and Cluster Systems“, Springer 2010)</li><li>• K.A. Berman, J.L. Paul: "Sequential and Parallel Algorithms", PWS Publishing Company, 1997</li><li>• B. Chapman, G. Jost, R. van der Pas: "Using OpenMP - Portable Shared Memory Parallel Programming", MIT Press, 2008</li><li>• W. Gropp, E. Lusk, und R. Thakur: "Using MPI-2: Advanced Features of the Message-Passing Interface", das Buch ist auch in deutscher ubersetzung erhaltlich.</li><li>• D. Kirk, W.-M. Hwu Programming Massively Parallel Processors</li></ul>								
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"><li>• 424201 Vorlesung High Performance Computing</li><li>• 424202 ubung High Performance Computing</li></ul>								
16. Abschatzung Arbeitsaufwand:	<table><tr><td>Prsenzzeit:</td><td>42 h</td></tr><tr><td>Selbststudiums- /</td><td>138 h</td></tr><tr><td>Nachbearbeitungszeit:</td><td></td></tr><tr><td>Summe:</td><td>180 h</td></tr></table>	Prsenzzeit:	42 h	Selbststudiums- /	138 h	Nachbearbeitungszeit:		Summe:	180 h
Prsenzzeit:	42 h								
Selbststudiums- /	138 h								
Nachbearbeitungszeit:									
Summe:	180 h								
17. Prfungsnummer/n und -name:	42421 High Performance Computing (PL), schriftlich oder mundlich, 90 Min., Gewichtung: 1.0								
18. Grundlage fur ... :									
19. Medienform:									
20. Angeboten von:	Simulation groer Systeme								

## Modul: 51720 IT-Strategy

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	Sven Lorenz		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:			
12. Lernziele:	Die Vorlesung fokussiert auf Management Strategien. Es wird erläutert, wie solche Strategien entwickelt und evaluiert werden.  Teilnehmer der Vorlesung verstehen die Bestandteile einer IT Strategie. Sie können eine IT Strategie ableiten und entwickeln, basierend auf dem aktuellen Status eines Unternehmens. Insbesondere wird verstanden, was unter den Begriffen und Konzepten IT Organisation, Sourcing Management, Architektur Management, Qualitäts- und Risk-Management und schliesslich IT Landschaften zu verstehen ist und wie man damit umgeht.		
13. Inhalt:	Über die Einstiegsfragestellung „Was ist ‚Strategie‘?“ wird erläutert, was eine Unternehmensstrategie und eine IT-Strategie ist, wobei sowohl die klassischen Ansätze als auch neue Sichtweisen vorgestellt werden. Im Schwerpunkt „Strategieentwicklung“ wird auf die Ableitung der IT-Strategie aus der Unternehmensstrategie eingegangen. Ein kanonisches Vorgehensmodell wird eingeführt und anhand von Unternehmensbeispielen illustriert. Der Schwerpunkt „IT-Strategie als Prozess“ beginnt mit der Einbettung der IT-Strategieaufgaben in die bekannten IT Prozessmodelle wie ITIL und CobiT. Im Rahmen eines verallgemeinerten IT-Prozessmodells werden die einzelnen IT-Strategieprozesse (IT-Organisationsentwicklung, IT-Sourcing-Strategie, IT-Architektur-Management, IT-Bebauungsplanung, IT-Qualitätsmanagement und IT-Risikomanagement) in der Folge detailliert erläutert. Dabei werden klassische und State-of-the-art Methoden und Werkzeuge zur Unterstützung der IT-Strategieprozesse vorgestellt. Exkurse in das IT-Portfoliomanagement und in IT-Kennzahlensysteme runden die Vorlesungsinhalte ab.		
14. Literatur:	<ul style="list-style-type: none"> <li>• Helmut Krcmar, „Informationsmanagement“, Springer, 2010</li> <li>• Jürgen Hofmann, Werner Schmitt, „Masterkurs IT-Management“, VIEWEG+TEUBNER, 2010W.</li> <li>• Brenner, A. Resch, V. Schulz, „Die Zukunft der IT in Unternehmen“, FAZ Buch, 2010</li> <li>• Martin Kütz, „Kennzahlen in der IT“, dpunkt-Verlag, 2007</li> </ul>		
15. Lehrveranstaltungen und -formen:	517201 Vorlesung mit Übungen IT-Strategie		
16. Abschätzung Arbeitsaufwand:	Lecture & exercises: 42 hours Self-study: 138 hours		

17. Prüfungsnummer/n und -name: 51721 IT-Strategy (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0

---

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von: Institut für Architektur von Anwendungssystemen

---

## Modul: 48500 Image Synthesis

2. Modulkürzel:	051903654	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Martin Fuchs		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Thomas Ertl</li> <li>• Daniel Weiskopf</li> <li>• Martin Fuchs</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	Modul 10060 Computergraphik		
12. Lernziele:	The students know the theoretical foundations of image synthesis and have practical expertise in programming of rendering systems. They know several approaches and algorithms for three-dimensional computer graphics, both for real-time and physically accurate rendering.		
13. Inhalt:	<p>The class covers physically based rendering techniques such as ray/path tracing and radiosity, computer graphics models for light transport and light/scene interaction, as well as numerical methods such as Monte Carlo integration and finite element methods which approximate solutions to the rendering equation. In addition, techniques which specifically employ modern graphics processing hardware are covered which approximate physically correct solutions in interactive application scenarios by means of rasterization and image-space rendering.</p> <p>Specifically, the class covers:</p> <ul style="list-style-type: none"> <li>• graphics hardware and rasterization APIs by example of OpenGL</li> <li>• textures and procedural models</li> <li>• shading and shadow computations in rasterization pipelines</li> <li>• scene graphs, culling and level-of-detail approaches</li> <li>• physically based rendering and photo-realistic image synthesis</li> <li>• local shading and material models, especially the BRDF</li> <li>• the rendering equation</li> <li>• ray tracing and Monte-Carlo approaches</li> <li>• global illumination simulation (especialy by means of radiosity, distriubtion ray tracing and path tracing)</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Andrew S. Glassner: Principles of Digital Image Synthesis, 1995J.</li> <li>• Foley, A. van Dam, S. Feiner, J. Hughes: Computer Graphics: Principle and Practice, 1990M.</li> <li>• Pharr, G. Humphreys: Physically Based Rendering, 2004</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 485001 Lecture Image Synthesis</li> <li>• 485002 Exercise Image Synthesis</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	



## Modul: 55610 Information Integration

2. Modulkürzel:	051210166	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Melanie Herschel		
9. Dozenten:	Melanie Herschel		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Lecture "Modellierung" or comparable course		
12. Lernziele:	Integrating heterogeneous, autonomous and structured data is essential in an interconnected world. This is the basis for information exchange and comprehensive search. The goal of this course is to provide an overview of challenges in information integration and to enable the students to assess available approaches and technologies.		
13. Inhalt:	Based on application scenarios from various organizations, we will discuss aspects of distribution, autonomy and heterogeneity. This helps us to organize the problem space and to compare possible architectures of integrated information systems. Heterogeneity is addressed by schema mappings between and data mappings. We will discuss how to establish such mappings and how to apply them in data transformation. As query processing in federated databases is based on these mappings as well, we will also learn the basics on these systems. Another focus of this course is on the pre-processing and integration of data. Starting with a discussion on information quality, we will look at the spectrum of erroneous data and approaches to data cleansing. State-of-the-art software for information integration will be presented, in particular as part of the exercises.		
14. Literatur:	Additional literature will be announced at the beginning of the lecture <ul style="list-style-type: none"> <li>• Ulf Leser, Felix Naumann: Informationsintegration: Architekturen und Methoden zur Integration verteilter und heterogener Datenquellen, dpunkt Verlag, 2006, ISBN 3898644006.</li> <li>• AnHai Doan, Alon Halevy, Zachary Ives: Principles of Data Integration, Morgan Kaufmann, 2012, ISBN 978-0-12-416044-6</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 556101 Vorlesung Information Integration</li> <li>• 556102 Übung Information Integration</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden  <b>Gesamt: 180 Stunden</b>		
17. Prüfungsnummer/n und -name:	55611 Information Integration (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			

19. Medienform:

20. Angeboten von:

Data Engineering

---

## Modul: 55630 Information Visualization and Visual Analytics

2. Modulkürzel:	051900099	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Daniel Weiskopf		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Steffen Koch</li> <li>• Thomas Ertl</li> <li>• Daniel Weiskopf</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	Basic Human Computer Interaction		
12. Lernziele:	Student gains expertise about fundamental concepts and techniques of information visualization and visual analytics. This includes algorithms and mathematical background, data structures and implementation aspects as well as practical experience with widely available visualization tools.		
13. Inhalt:	<p>Topics covered in this course:</p> <ul style="list-style-type: none"> <li>- Perception and Cognition</li> <li>- Graphs and Networks</li> <li>- Hierarchies and Trees</li> <li>- Multi-dimensional and high-dimensional data visualization</li> <li>- Time series visualization</li> <li>- Visual Analytics</li> <li>- Software Visualization</li> <li>- Geospatial visualization</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Colin Ware. Visual Thinking for Design</li> <li>• Colin Ware. Information Visualization. Perception for Design</li> <li>• Edward Tufte. The Visual Display of Quantitative Information</li> <li>• Robert Spence. Design for Interaction</li> <li>• Jim Thomas. Illuminating the Path</li> </ul>		
15. Lehrveranstaltungen und -formen:	556301 Vorlesung und Übung Informationsvisualisierung		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit: 42 Stunden          Selbststudium: 138 Stunden</p> <p><b>Gesamt: 180 Stunden</b></p>		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 55631 Information Visualization and Visual Analytics (PL), schriftlich, eventuell mündlich, 120 Min., Gewichtung: 1.0</li> <li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich, Erfolgreiche Übungsteilnahmen / excercises passed</li> </ul>		

18. Grundlage für ... :

---

19. Medienform: Video projector, blackboard, exercises using PCs

---

20. Angeboten von: Institut für Visualisierung und Interaktive Systeme

---

## Modul: 60120 Interaktive Systeme

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes Semester
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Univ.-Prof. Albrecht Schmidt		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Albrecht Schmidt</li> <li>• Niels Henze</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	Vorlesung Mensch-Computer Interaktion		
12. Lernziele:	Die Studierenden lernen, wie interaktive Systeme entwickelt werden. Sie verstehen den Entwicklungsprozess und können interaktive Systeme für spezifische Plattform entwickeln.		
13. Inhalt:			
14. Literatur:			
15. Lehrveranstaltungen und -formen:	601201 Fachpraktikum Interaktive Systeme		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit: 56 Stunden</p> <p>Selbststudium: 124 Stunden</p>		
17. Prüfungsnummer/n und -name:	60121 Interaktive Systeme (LBP), Sonstiges, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

## Modul: 29460 Kryptographische Verfahren

2. Modulkürzel:	050420110	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Apl. Prof. Ulrich Hertrampf		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Ulrich Hertrampf</li> <li>• Volker Diekert</li> <li>• Stefan Funke</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	Theorie-Vorlesungen des Bachelor-Studiums		
12. Lernziele:	Die Studierenden kennen die wichtigsten Sätze der Kryptographie. Sie können klassische und moderne Verschlüsselungsverfahren anwenden und die Sicherheit dieser Verfahren beurteilen und einstufen.		
13. Inhalt:	<p>Moderne Verfahren der einstigen "Geheimwissenschaft" Kryptographie werden eingeführt. Die Veranstaltung stellt Methoden zur Erzeugung elektronischer Unterschriften und zur Identifikation von Benutzern vor, die als notwendige Voraussetzungen für elektronische Wahlen oder anonymes elektronisches Bargeld gelten. Es werden neben klassischen, symmetrischen Verschlüsselungsverfahren aktuelle asymmetrische Verfahren behandelt. Eine wichtige Rolle spielen Protokolle, die aufbauend auf kryptographischen Verfahren die erwähnten Aufgaben lösen.</p>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Bruce Schneier, Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C, 1996</li> <li>• Douglas Robert Stinson, Cryptography: Theory and Practice, 1995</li> <li>• Friedrich Ludwig Bauer, Entzifferte Geheimnisse: Methoden und Maximen der Kryptologie, 1995</li> <li>• Johannes Buchmann, Einführung in die Kryptographie, 1999</li> </ul>		
15. Lehrveranstaltungen und -formen:	294601 Vorlesung mit Übungen Kryptographische Verfahren		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 29461 Kryptographische Verfahren (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0</li> <li>• V Vorleistung (USL-V), schriftlich oder mündlich</li> </ul>		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Theoretische Informatik		

## Modul: 29480 Loose Coupling and Message Based Applications

2. Modulkürzel:	052010003	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Frank Leymann</li> <li>• Dimka Karastoyanova</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:			
12. Lernziele:	Understand the problem of application integration and the fundamental concept of loose coupling. The pros and cons of messaging are clear, and the architecture of Message Oriented Middleware is understood. Key patterns of using messaging to solve (enterprise) application integration problems are understood.		
13. Inhalt:	<p>Messaging is a cornerstone of the integration of heterogeneous applications inside and among enterprises. Applications that need to share data synchronously or asynchronously with each other can be made to interoperate by means of the feature-rich Message-Oriented Middleware (MOM) that has grown ubiquitous in enterprises. During this course we treat the approaches and challenges of application integration through messaging. At first, we will address concepts such as (a-)synchronous messaging and the different messaging styles, e.g. point-to-point and publish-subscribe, that are the foundation of message-based application integration. Later in the course we will take an in-depth look at the mechanics and architecture of MOM, in particular of the Java Messaging Service (JMS), which will also be used in examples and exercises. Throughout the course we will discuss and apply extensively Enterprise Application Integration (EAI) patterns. Especially, endpoint patterns, routing patterns, transformation patterns, messaging patterns, channel patterns, and management patterns will be presented; the composability of these patterns will be explained.</p>		
14. Literatur:	<p>G. Hohpe and B. Woolf: "Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions." Addison-Wesley Professional, ISBN-13: 978-0321200686. October 2003.</p> <p>M. Hapner et al: "Java Messaging Service API Tutorial &amp; Reference". Addison-Wesley 2001.</p>		
15. Lehrveranstaltungen und -formen:	294801 Vorlesung mit Übungen Lose Kopplung & Message-basierte Integration		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit: 42 Stunden          Nachbearbeitungszeit: 138 Stunden</p>		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 29481 Loose Coupling and Message Based Applications (PL), schriftlich oder mündlich, 60 Min., Gewichtung: 1.0, mündl. Prüfungsdauer: 30 Minuten</li> </ul>		

• V Vorleistung (USL-V), Sonstiges

---

18. Grundlage für ... :

---

19. Medienform: Lecture and accompanying exercises

---

20. Angeboten von: Architektur von Anwendungssystemen

---

## Modul: 29470 Machine Learning

2. Modulkürzel:	051200112	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Marc Toussaint		
9. Dozenten:	Marc Toussaint		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	Solid knowledge in Linear Algebra, probability theory and optimization. Fluency in at least one programming language.		
12. Lernziele:	<p>Students will acquire an in depth understanding of Machine Learning methods. The concepts and formalisms of Machine Learning are understood as generic approach to a variety of disciplines, including image processing, robotics, computational linguistics and software engineering. This course will enable students to formalize problems from such disciplines in terms of probabilistic models and the derive respective learning and inference algorithms.</p>		
13. Inhalt:	<p>Exploiting large-scale data is a central challenge of our time. Machine Learning is the core discipline to address this challenge, aiming to extract useful models and structure from data. Studying Machine Learning is motivated in multiple ways: 1) as the basis of commercial data mining (Google, Amazon, Picasa, etc), 2) a core methodological tool for data analysis in all sciences (vision, linguistics, software engineering, but also biology, physics, neuroscience, etc) and finally, 3) as a core foundation of autonomous intelligent systems (which is my personal motivation for research in Machine Learning).</p> <p>This lecture introduces to modern methods in Machine Learning, including discriminative as well as probabilistic generative models. A preliminary outline of topics is:</p> <ul style="list-style-type: none"> <li>• motivation and history</li> <li>• probabilistic modeling and inference</li> <li>• regression and classification methods (kernel methods, Gaussian Processes, Bayesian kernel logistic regression, relations)</li> <li>• discriminative learning (logistic regression, Conditional Random Fields)</li> <li>• feature selection</li> <li>• boosting and ensemble learning</li> <li>• representation learning and embedding (kernel PCA and derivatives, deep learning)</li> <li>• graphical models</li> <li>• inference in graphical models (MCMC, message passing, variational)</li> <li>• learning in graphical models</li> <li>• structure learning and model selection</li> <li>• relational learning</li> </ul> <p>Please also refer to the course web page: <a href="http://ipvs.informatik.uni-stuttgart.de/mlr/marc/teaching/13-MachineLearning/">http://ipvs.informatik.uni-stuttgart.de/mlr/marc/teaching/13-MachineLearning/</a></p>		

14. Literatur:
- [1] *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* by Trevor Hastie, Robert Tibshirani and Jerome Friedman. Springer, Second Edition, 2009.  
full online version available: <http://www-stat.stanford.edu/~tibs/ElemStatLearn/>  
(recommended: read introductory chapter)
  - [2] *Pattern Recognition and Machine Learning* by Bishop, C. M.. Springer 2006.  
online: <http://research.microsoft.com/en-us/um/people/cmbishop/prml/>  
(especially chapter 8, which is fully online)

---

15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"><li>• 294701 Lecture Machine Learning</li><li>• 294702 Exercise Machine Learning</li></ul>
16. Abschätzung Arbeitsaufwand:	Presence time: 42 hours Self study: 138 hours Sum: 180 hours
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"><li>• 29471 Machine Learning (PL), schriftlich, eventuell mündlich, 120 Min., Gewichtung: 1.0</li><li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich</li></ul>
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Institut für Parallele und Verteilte Systeme

---

## Modul: 29640 Mikrocontroller

2. Modulkürzel:	051230115	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	<p>Knowledge of at least one programming language and knowledge in the field of computer science or similar subjects.</p> <p>Kenntnisse in mindestens einer Programmiersprache und in mindestens einem Fach aus dem Bereich dem Bereich der Technischen Informatik oder ähnlichen Fächern.</p>		
12. Lernziele:	<p>Students are able to master the practical programming of microcontrollers and are familiar with classical architectures.</p> <p>Historical Overview          Microcontroller architectures          Applications of microcontrollers          Instruction set classic microcontroller          Assembly language programming of microcontrollers          C programming for microcontrollers</p> <p>Studierende beherrschen die praktische Programmierung von Mikrocontrollern und kennen klassische Architekturen.</p> <ul style="list-style-type: none"> <li>• Historische Übersicht</li> <li>• Mikrocontroller-Architekturen</li> <li>• Einsatzgebiete von Mikrocontrollern</li> <li>• Befehlssatz klassischer Microcontroller</li> <li>• Assembler-Programmierung von Mikrocontrollern</li> <li>• C-Programmierung von Mikrocontrollern</li> </ul>		
13. Inhalt:	<p>Microcontrollers (also called <math>\mu</math>Controller, <math>\mu</math>C, MCU) are IC's that combine at least peripheral functions on a single chip. In many cases, the working and programming memory is also partially or completely on the same chip . A microcontroller is practically a one-chip computer system. The number of built-in microcontroller exceeds by far the number of microprocessors . A microcontroller is often part of an embedded system in devices of everyday life like washing machines, smart cards (money, telephone cards), consumer electronics (VCRs, disc players, radios, televisions, remote controls), office electronics, motor vehicles (ECU for ABS, airbag, engine, instrument cluster, ESP, etc.), mobile phones and even in clocks and watches. In addition they are found on virtually all computer peripherals including</p>		

keyboards, mouse, printers, monitors, scanners etc.

Microcontrollers are adapted to performance and respective features of the application. Therefore they have significant advantages in cost and power consumption compared with normal computers.

Small microcontrollers are available in high numbers for less than 1\$.

Als Microcontroller (auch  $\mu$ Controller,  $\mu$ C, MCU ) werden ICs bezeichnet, die mit dem Prozessor mindestens Peripheriefunktionen auf einem Chip vereinen. In vielen Fällen befindet sich der Arbeits- und Programmspeicher ebenfalls teilweise oder komplett auf dem gleichen Chip. Ein Mikrocontroller ist praktisch ein Ein-Chip-Computersystem. Die Anzahl der verbauten Mikrocontroller überschreitet bei weitem die Zahl der Mikroprozessoren.

Der Mikrocontroller tritt in Gestalt von eingebetteten Systemen im Alltag oft unbemerkt in technischen Gebrauchsartikeln auf, zum Beispiel in Waschmaschinen, Chipkarten (Geld-, Telefonkarten), Unterhaltungselektronik (Videorekordern, CD-/DVD-Playern, Radios, Fernsehgeräten, Fernbedienungen), Büroelektronik, Kraftfahrzeugen (Steuergeräte für z.B. ABS, Airbag, Motor, Kombiinstrument, ESP usw.), Mobiltelefonen und sogar in Uhren und Armbanduhren. Darüber hinaus sind sie in praktisch allen Computer-Peripheriegeräten enthalten (Tastatur, Maus, Drucker, Monitor, Scanner uvm.).

Mikrocontroller sind in Leistung und Ausstattung auf die jeweilige Anwendung angepasst. Daher haben sie gegenüber normalen Computern deutliche Vorteile bei den Kosten und der Leistungsaufnahme. Kleine Mikrocontroller sind in höheren Stückzahlen für deutlich unter 1\$, - verfügbar.

Aus <http://de.wikipedia.org/wiki/Mikrocontroller>

---

14. Literatur:	<ul style="list-style-type: none"> <li>• Jörg Wiegelmann, Softwareentwicklung in C für Mikroprozessoren und Mikrocontroller: C- Programmierung für Embedded-Systeme, 2009</li> </ul> <p>More literature is named in the lecture</p>
<hr/>	
15. Lehrveranstaltungen und -formen:	296401 Vorlesung mit Übung Mikrocontroller
<hr/>	
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit: 42 Stunden</p> <p>Selbststudium: 138 Stunden</p> <p><b>Gesamt: 180 Stunden</b></p>
<hr/>	
17. Prüfungsnummer/n und -name:	29641 Mikrocontroller (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0, Schriftliche Prüfung von 120 Min. oder mündlichen Prüfung von 30 Min.
<hr/>	
18. Grundlage für ... :	
<hr/>	
19. Medienform:	
<hr/>	
20. Angeboten von:	
<hr/>	

## Modul: 29720 Mobile Computing

2. Modulkürzel:	051200166	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Kurt Rothermel		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Kurt Rothermel</li> <li>• Frank Dürr</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	Rechnernetze		
12. Lernziele:	<p>The knowledge that has been acquired in the course "Computer Networks I" regarding concepts, protocols, and technologies of computer networks , will be extended to mobile devices and wireless communication systems and procedures. The objective of this lecture is to understand problems that might occur in the usage of mobile devices as well as to obtain knowledge to develop solutions for these problems and to communicate with experts. The Participants will learn about advantages and the disadvantages of specific wireless communication technologies for mobile devices and will be able to use appropriate protocols for the applications or modify them as needed. The exercises are used to provide practical experience in programming, analysis, performance evaluation of mobile and wireless communication systems as well as the expertise in the usage of appropriate tools.</p>		
13. Inhalt:	<ol style="list-style-type: none"> <li>1. Fundamentals of wireless data transmission</li> <li>2. Media access for wireless networks</li> <li>3. Location Management</li> <li>4. Wireless Wide Area Networks</li> <li>5. Wireless networks (local/personal)</li> <li>6. Ad-hoc Networks: Exchange, Location administration</li> <li>7. Mobility in IP-networks</li> <li>8. Transport layer protocols for mobile systems</li> <li>9. Location of services</li> <li>10. Mobile data access</li> <li>11. Introduction</li> <li>12. Wireless data transmission</li> <li>13. Location Management</li> <li>14. Wireless</li> <li>15. Telephone communication systems : GSM, GPRS,UMTS</li> <li>16. Wireless networks (local/personal): 802.11, Bluetooth</li> <li>17. Ad-hoc Networks: Routing, Location Management</li> <li>18. Internetworking: Mobile IP, Cellular IP</li> <li>19. Transport layers for mobile systems</li> <li>20. Location of services : Problem, JINI, UpnP</li> <li>21. Mobile data access: Broadcast Scheduling, Hoarding</li> </ol>		
14. Literatur:	<p>Charles E. Perkins: Mobile IP: Design Principles and Practices. 1997          James D. Solomon: Mobile IP: The Internet Unplugged. 1998          Jochen Schiller: Mobile Communications. 2000</p>		

Jörg Roth: Mobile Computing: Grundlagen, Technik und Konzepte. 2002  
Kian-Lee Tan, Beng-Chin Ooi: Data Dissemination in Wireless Computing Envi-ronments. 2000  
Tomasz Imielinski, Henry F. Korth (ed.): Mobile Computing. 1996

---

15. Lehrveranstaltungen und -formen:	297201 Vorlesung mit Übung Mobile Computing
16. Abschätzung Arbeitsaufwand:	Mobile Computing Vorlesung - Präsenzzeit: 21 Stunden - Selbststudium: 69 Stunden Mobile Computing Übungen - Präsenzzeit: 21 Stunden - Selbststudium: 69 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"><li>• 29721 Mobile Computing (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0, Prüfungsdauer: 90 min schriftlich oder 30 min mündlich Exam duration: 90 min written exam or 30 min oral exam</li><li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich</li></ul>
18. Grundlage für ... :	
19. Medienform:	Folien, Tafel
20. Angeboten von:	Verteilte Systeme

---

## Modul: 10120 Modellbildung und Simulation

2. Modulkürzel:	051240010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Dirk Pflüger		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Dirk Pflüger</li> <li>• Stefan Zimmer</li> <li>• Miriam Mehl</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 10190 Mathematik für Informatiker und Softwaretechniker</li> <li>• Modul 10240 Numerische und Stochastische Grundlagen der Informatik</li> </ul>		
12. Lernziele:	Beherrschung des grundsätzlichen Vorgehens in der Modellbildung. Kenntnis einer Auswahl diskreter und kontinuierlicher Modelle und entsprechender Simulationsmethoden. Fähigkeit, mit den erlernten Kenntnissen selbständig numerische Methoden problemorientiert um- und einzusetzen.		
13. Inhalt:	Diese Vorlesung bietet eine Einführung in die Grundlagen der Modellbildung und Simulation mit dem Ziel der Vorbereitung auf weiterführende Vorlesungen in diesem Bereich. Da Simulationsmethoden oft für viele verschiedene Problemklassen einsetzbar sind, ist die Vorlesung methodisch strukturiert. Den Hauptteil der Vorlesung bilden hierbei diskrete Modelle sowie deren Behandlung, aber auch kontinuierliche Modelle werden ergänzend gestreift. Ob diskrete Ereignissimulation, spieltheoretische Ansätze, Zelluläre Automaten, Räuber-Beute Modelle oder Fuzzy-Mengen: die verschiedenen Modellierungsansätze sind so vielfältig wie die Problemstellungen, auf die sie angewendet werden. Verkehrssimulation, Populationswachstum, Wahlen oder Regelung sind nur einige der Anwendungsbereiche aus den Natur- und Ingenieurwissenschaften.		
14. Literatur:	<ul style="list-style-type: none"> <li>• Modellbildung und Simulation - Eine anwendungsorientierte Einführung; Bungartz, H.-J., Zimmer, S., Buchholz, M., Pflüger, D., Springer Verlag, eXamen.press, 2013, ISBN 978-3-642-38656-6</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 101201 Vorlesung Modellbildung und Simulation</li> <li>• 101202 Übung Modellbildung und Simulation</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	10121 Modellbildung und Simulation (PL), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			

20. Angeboten von: Simulation großer Systeme

---

## Modul: 29730 Modelling, Simulation, and Specification

2. Modulkürzel:	051711020	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Martin Radetzki		
9. Dozenten:	Martin Radetzki		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:			
12. Lernziele:	Master-level understanding of fundamental models of computation and their simulation, ability to apply them to embedded systems specification.		
13. Inhalt:	Given the complexity and implementation cost of contemporary electronic systems, it is essential to specify their intended functionality before elaborating the implementation. This course focuses on the model-based and executable specification of embedded systems and covers the following topics: <ul style="list-style-type: none"> <li>• Hierarchical concurrent state machine models;</li> <li>• Kahn process networks, synchronous data flow networks;</li> <li>• Models of computation;</li> <li>• Tagged signal model;</li> <li>• Specification of timing, concurrency, and non-functional aspects;</li> <li>• Event-driven simulation;</li> <li>• Statically scheduled simulation;</li> <li>• Parallel simulation techniques</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Lecture Notes “Modelling, Simulation, and Specification”.</li> <li>• Jantsch: Modeling Embedded Systems and SoCs Concurrency and Time in Models of Computation. Morgan Kaufman Publishers, 2004.</li> <li>• Black, D.; Donovan,D.: SystemC from the Ground Up. Kluwer Academic Publishers, 2004.</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 297301 Vorlesung Modelling, Simulation, and Specification</li> <li>• 297302 Übung Modelling, Simulation, and Specification</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden  <b>Summe: 180 Stunden</b>		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 29731 Modelling, Simulation, and Specification (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0</li> <li>• V Vorleistung (USL-V), schriftlich oder mündlich, Als Zulassungsvoraussetzung zur Klausur ist die folgende</li> </ul>		

Vorleistung zu erbringen: Teilnahme an den Übungen,  
Präsentation der Lösung wenigstens einer Aufgabe.

---

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von: Eingebettete Systeme (Embedded Systems Engineering)

---

## Modul: 55650 Multimodal Interaction for Ubiquitous Computers

2. Modulkürzel:	051900033	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Albrecht Schmidt		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Albrecht Schmidt</li> <li>• Niels Henze</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	Basics of human computer interaction		
12. Lernziele:	Broad understanding for methods and concepts of multimodal interactions of personal computers, in particular for mobile systems, vehicles, tedious devices and environments.		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Interaction with mobile phones</li> <li>• User interfaces for vehicles</li> <li>• Interaction with intelligent environments</li> <li>• Interactive interfaces and gestures</li> <li>• Tangible user interfaces</li> <li>• Speech input and output</li> <li>• Camera-based interaction</li> <li>• Physiological sensors as interfaces between human and computer</li> <li>• Activities, context and emotions as input</li> <li>• Methods and techniques for designing user interfaces</li> <li>• Approaches for evaluating user interfaces</li> </ul>		
14. Literatur:			
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 556501 Lecture Multimodal Interaction for Ubiquitous Computers</li> <li>• 556502 Exercise Multimodal Interaction for Ubiquitous Computers</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	55651 Multimodal Interaction for Ubiquitous Computers (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme		

## Modul: 42460 Numerische Simulation

2. Modulkürzel:	051240060	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Dirk Pflüger		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Dirk Pflüger</li> <li>• Stefan Zimmer</li> <li>• Miriam Mehl</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 10190 Mathematik für Informatiker und Softwaretechniker und</li> <li>• Modul 10240 Numerische und Stochastische Grundlagen der Informatik bzw.</li> <li>• Modul 41590 Einführung in die Numerik und Stochastik für Softwaretechniker</li> <li>• Modul 42410 Grundlagen des wissenschaftlichen Rechnens</li> </ul>		
12. Lernziele:	Fähigkeit zur Implementierung numerischer Methoden und Entwicklung und Umsetzung geeigneter Datenstrukturen.		
13. Inhalt:	Strukturmechanik, Strömungsmechanik, Finite Elemente, Finite Differenzen sowie praktische Aspekte der effizienten und parallelen Umsetzung auf Rechnern.		
14. Literatur:	<ul style="list-style-type: none"> <li>• Griebel, Dornseifer, Neunhoeffler: Numerical simulation in fluid dynamics : a practical introduction; SIAM, 1998 / Numerische Simulation in der Strömungsmechanik; Vieweg 1995</li> <li>• Griebel, Knapek, Zumbusch, Caglar: Numerische Simulation in der Moleküldynamik : Numerik, Algorithmen, Parallelisierung, Anwendungen; Springer 2004</li> <li>• Braess: Finite Elemente : Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie; Springer, 2007</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 424601 Vorlesung Numerische Simulation</li> <li>• 424602 Übung Numerische Simulation</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	42461 Numerische Simulation (LBP), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Simulation großer Systeme		

## Modul: 40680 Optimization

2. Modulkürzel:	051200113	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Marc Toussaint		
9. Dozenten:	Marc Toussaint		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Solid basic knowledge in linear algebra and analysis. Basic programming skills.		
12. Lernziele:	Students will learn to identify, mathematically formalize, and derive algorithmic solutions to optimization problems as they occur in nearly all disciplines, e.g. Machine Learning, Combinatorial Optimization, Computer Vision, Robotics, Simulation. The focus will be on continuous optimization problems (including as they arise from relaxations of discrete problems), including convex problems, quadratic & linear programming, but also non-linear black-box problems. The goal is to give an overview of the various approaches and mathematical formulations and practical experience with the basic paradigms.		
13. Inhalt:	<p>Optimization is one of the most fundamental tools of modern sciences. Many phenomena -- be it in computer science, artificial intelligence, logistics, physics, finance, or even psychology and neuroscience -- are typically described in terms of optimality principles. The reason is that it is often easier to describe or design an optimality principle or cost function rather than the system itself. However, if systems are described in terms of optimality principles, the computational problem of optimization becomes central to all these sciences.</p> <p>This lecture aims give an overview and introduction to various approaches to optimization together with practical experience in the exercises. The focus will be on continuous optimization problems and we will cover methods ranging from standard convex optimization and gradient methods to non-linear black box problems (evolutionary algorithms) and optimal global optimization. Students will learn to identify, mathematically formalize, and derive algorithmic solutions to optimization problems as they occur in nearly all disciplines. A preliminary list of topics is:</p> <ul style="list-style-type: none"> <li>• gradient methods, log-barrier, conjugate gradients, Rprop</li> <li>• constraints, KKT, primal/dual</li> <li>• Linear Programming, simplex algorithm</li> <li>• (sequential) Quadratic Programming</li> <li>• Markov Chain Monte Carlo methods</li> <li>• 2nd order methods, (Gauss-)Newton, (L)BFGS</li> <li>• blackbox stochastic search, including a discussion of evolutionary algorithms</li> </ul> <p>Please also refer to the course web page: <a href="http://ipvs.informatik.uni-stuttgart.de/mlr/marc/teaching/13-Optimization/">http://ipvs.informatik.uni-stuttgart.de/mlr/marc/teaching/13-Optimization/</a></p>		

14. Literatur:

---

15. Lehrveranstaltungen und -formen: 406801 Vorlesung mit Übungen Optimization

---

16. Abschätzung Arbeitsaufwand: Presence time: 42 hours  
Self study: 138 hours  
Sum: 180 hours

---

17. Prüfungsnummer/n und -name: 40681 Optimization (PL), schriftlich, eventuell mündlich, 120 Min.,  
Gewichtung: 1.0

---

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von: Institut für Parallele und Verteilte Systeme

---

## Modul: 56790 Parallele Numerik

2. Modulkürzel:	051240080	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Miriam Mehl		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Miriam Mehl</li> <li>• Dirk Pflüger</li> <li>• Stefan Zimmer</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 6. Semester  → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT  →</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester  → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 41590 Einführung in die Numerik und Stochastik für Softwaretechniker <b>oder</b></li> <li>• Modul 10240 Numerische und Stochastische Grundlagen</li> </ul>		
12. Lernziele:	<p>Die Studenten kennen die wesentlichen parallelisierbaren Algorithmen für zentrale numerische Problemstellungen. Sie erkennen Parallelisierungshindernisse in bekannten und neuen numerischen Algorithmen, können die zu erwartende Skalierbarkeit abschätzen und sind in der Lage, Algorithmen so zu modifizieren, dass die parallele Effizienz erhöht wird ohne wichtige numerische Eigenschaften wie Stabilität und Komplexität zu verlieren.</p>		
13. Inhalt:	<ul style="list-style-type: none"> <li>• parallele Matrix- und Vektoroperationen</li> <li>• parallele Fouriertransformation</li> <li>• parallele QR Zerlegung und Least Squares Probleme</li> <li>• parallele iterative Gleichungssystemlöser</li> <li>• parallele Eigenwert- und Eigenvektorberechnung</li> <li>• parallele Zeitschrittverfahren</li> <li>• parallele Algorithmen für Teilchenwechselwirkungen</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Introduction to High Performance Scientific Computing (Eijkhout, Chow, van de Geijn) (download at <a href="http://www.lulu.com/shop/victor-eijkhout/introduction-to-high-performance-scientific-computing/paperback/product-21431780.html;jsessionid=CF30CC0B65B0F349BFBD206D406F8">http://www.lulu.com/shop/victor-eijkhout/introduction-to-high-performance-scientific-computing/paperback/product-21431780.html;jsessionid=CF30CC0B65B0F349BFBD206D406F8</a>)</li> <li>• Numerical Linear Algebra for High-Performance Computers (Dongarra, Duff, Sorensen, van der Vorst)</li> <li>• Parallel Algorithms for Matrix Computations (Gallivan, Heath, Ng, Ortega,...)</li> <li>• A User's Guide to MPI (Pacheco)</li> <li>• Iterative Methods for Sparse Linear Systems (Saad)</li> <li>• Lösung linearer Gleichungssysteme auf Parallelrechnern (Frommer)</li> <li>• M. Griebel, S. Knapek, G. Zumbusch, and A. Caglar. Numerische Simulation in der Molekulardynamik. Springer, 2004.</li> <li>• D. Frenkel and B. Smith. Understanding Molecular Simulation from Algorithms to Applications. Academic Press (2nd ed.), 2002.</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 567901 Vorlesung Parallele Numerik</li> <li>• 567902 Übung Parallele Numerik</li> </ul>		

16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h
	Selbststudiums- /	138 h
	Nachbearbeitungszeit:	
	Summe:	180 h
<hr/>		
17. Prüfungsnummer/n und -name:	56791 Parallele Numerik (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0	
<hr/>		
18. Grundlage für ... :		
<hr/>		
19. Medienform:		
<hr/>		
20. Angeboten von:	Simulation großer Systeme	
<hr/>		

## Modul: 29650 Parallele Programmierung

2. Modulkürzel:	051230130	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Kenntnisse in mindestens einer Programmiersprache.Kenntnisse in mindestens einem Fach der Technischen Informatik odereinem ähnlichen Fach.		
12. Lernziele:	Studierende beherrschen die Programmierung von Multi-Core Prozessoren und paralleler Rechner bzw. Computing-Systemen.		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Grundlegende Parallelisierungsansätze: Parallelisierung durch Datenzerlegung, parallele lineare Algebra, etc.</li> <li>• Message Passing Interface</li> <li>• Open MP</li> <li>• C-Programmierung für FPGAs</li> <li>• Graphische Programmierung</li> <li>• GPU-Programmierung</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Thomas Rauber und Gundula Rürger, Multicore: Parallele Programmierung (Informatik Im Fokus), 2007</li> <li>• More literature is named in the lecture</li> </ul>		
15. Lehrveranstaltungen und -formen:	296501 Vorlesung mit Übung Parallele Programmierung		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	29651 Parallele Programmierung (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

## Modul: 10250 Parallele Systeme

2. Modulkürzel:	051200065	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Erfahrungen aus dem Bereich Technische Informatik		
12. Lernziele:	Grundlegende Kenntnisse im Bereich paralleler Systeme, z.B. Multi-Core CPUs und deren Programmierung.		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Die Entwicklung vom klassischen Mikroprozessor zur Multi-Core CPU Programmierung paralleler Rechnersysteme</li> <li>• Systolische Arrays, massiv parallele Systeme</li> <li>• Parallele Systeme aus verschiedenen Anwendungsdomänen: ausgewählte Fallbeispiele</li> </ul>		
14. Literatur:	Wird in der Lehrveranstaltung bekannt gegeben.		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 102501 Vorlesung Parallele Systeme</li> <li>• 102502 Übung Parallele Systeme</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 Stunden	
	Nachbearbeitungszeit:	138 Stunden	
	<b>Gesamt:</b>	<b>180 Stunden</b>	
17. Prüfungsnummer/n und -name:	10251 Parallele Systeme (LBP), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

## Modul: 48550 Practical Course Information Systems

2. Modulkürzel:	051200135	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Bernhard Mitschang		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Bernhard Mitschang</li> <li>• Holger Schwarz</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 3. Semester</p> <p>B.Sc. Softwaretechnik, PO 2012, 3. Semester</p> <p style="padding-left: 20px;">→ Ergänzungsmodule --&gt;Wahlmodule aus Master SWT</p> <p style="padding-left: 20px;">→</p> <p>B.Sc. Softwaretechnik, PO 2012, 3. Semester</p> <p style="padding-left: 20px;">→ Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	Basic knowledge on database systems, information systems and programming languages		
12. Lernziele:	Students get hands-on experience with state-of-the-art information systems. Students learn how to use these systems to address typical tasks in information processing. Based on this practical experience, they will also be able to assess available technologies and systems for various application areas.		
13. Inhalt:	The focus of this course is on the design and implementation of database-oriented applications. This includes core database technology as well as middleware and web technology.		
14. Literatur:	Will be announced at the beginning of the course		
15. Lehrveranstaltungen und -formen:	485501 Informationssystem-Fachpraktikum		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 60 Stunden Selbststudium: 120 Stunden		
17. Prüfungsnummer/n und -name:	48551 Practical Course Information Systems (LBP), schriftlich oder mündlich, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Datenbanken und Informationssysteme		

## Modul: 48570 Practical Course Visual Computing

2. Modulkürzel:	051900111	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Thomas Ertl		
9. Dozenten:	Thomas Ertl		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Basics of Computer Graphics		
12. Lernziele:	During this practical course, students will learn about approaches to rendering and visual computing technologies and will know how to implement these. They will learn about polygon based approach as well as volume rendering approaches. The students will learn, how to proceed a small project on their own (independently).		
13. Inhalt:	<ul style="list-style-type: none"> <li>• OpenGL</li> <li>• Qt-Framework</li> <li>• Raytracing</li> <li>• Volume Rendering</li> <li>• Independent Project</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• OpenGL Programming Guide - Third Edition (OpenGL 1.2) , Masonn Woo, Jackie Neider, Tom Davis, Dave Shreiner, Addison Wesley, 1999</li> <li>• Programming with Qt - First Edition, Matthias Kalle Dalheimer, O'Reilly, 1999</li> <li>• An Introduction to Ray Tracing, Andrew S. Glassner, Academic Press, 1989</li> <li>• Computer Graphics - Principle and Practice - Second Edition, Foley, van Dam, Feiner, Huges, Addison Wesley, 1990</li> </ul>		
15. Lehrveranstaltungen und -formen:	485701 Lab Practical Course Visual Computing		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	48571 Practical Course Visual Computing (LBP), schriftlich oder mündlich, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme		

## Modul: 29660 Programmanalysen und Compilerbau

2. Modulkürzel:	051510311	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Erhard Plödereder		
9. Dozenten:	Erhard Plödereder		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	Kenntnisse, die in etwa den Inhalten des Moduls 10150 - Grundlagen des Compilerbaus und der Programmiersprachen - des Bachelor-Studiums entsprechen, sind dringend empfohlen.		
12. Lernziele:	<p>Die Studierenden haben grundlegende Kenntnisse über die typischen in Compilern und anderen statischen Programmanalysen verwandten Verfahren erworben, sowohl in Bezug auf Basisanalysen (Kontroll- und Datenflussanalysen) als auch auf weitergehende, zielgerichteten Analysen wie Zeigeranalysen, Abhängigkeitsanalysen oder Slicing. Speziell lernen sie eine Reihe von Codeoptimierungen im Compiler kennen, aber auch diverse Globalanalysen, wie sie in Werkzeugen zur Fehlersuche, zum Reengineering oder zu Architekturanalysen nötig sind.</p>		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Attributgrammatiken (Wiederholung)</li> <li>• Programmanalysen und -Optimierung (Schwerpunkt)</li> <li>• klassische Optimierungen</li> <li>• Lokale und globale Kontrollflussanalyse</li> <li>• Lokale und globale Datenflussanalysen</li> <li>• Dominatoren, Dominatorgrenzen, Kontrollstrukturanalysen</li> <li>• Zeigeranalysen</li> <li>• Seiteneffekt-Analyse</li> <li>• Datenabhängigkeiten, Konfliktanalysen und Registervergabe</li> <li>• SSA-Form und ihre Berechnung</li> <li>• Code-Erzeugung</li> <li>• Implementierung von OOP</li> <li>• Das Laufzeitsystem</li> <li>• Separate Übersetzung</li> <li>• Slicing</li> <li>• Mustersuchen und Klonerkennung</li> <li>• Begriffsanalyse und ihre Anwendungen</li> </ul> <p>Orthogonal zu den jeweiligen Analyseverfahren werden die Verwendungen in Codeoptimierung und in Programmanalysen anderer Werkzeuge des Software Engineering aufgezeigt.</p>		
14. Literatur:	<ul style="list-style-type: none"> <li>• A. V. Aho, M.S. Lam, R. Sethi, J. D. Ullman: Compilers - Principles, Techniques, and Tools, Addison Wesley Verlag (2007)</li> <li>• Morgan, Robert, Building an Optimizing Compiler, 1998</li> <li>• Muchnick, Steven S., Advanced Compiler Design and Implementation, 1997</li> <li>• Wilhelm, Maurer: Übersetzerbau, 2. Auflage, Springer Verlag (1997)</li> <li>• Uwe Kastens: Übersetzerbau, Oldenbourg Verlag (1990)</li> </ul>		

15. Lehrveranstaltungen und -formen:	296601	Vorlesung mit Übung	Programmanalysen und Compilerbau
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:		42 h
	Selbststudiums- /		138 h
	Nachbearbeitungszeit:		
	Summe:		180 h
17. Prüfungsnummer/n und -name:	29661	Programmanalysen und Compilerbau (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0	
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Softwaretechnologie		

## Modul: 51740 Quantencomputing

2. Modulkürzel:	050420210	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Apl. Prof. Ulrich Hertrampf		
9. Dozenten:	Ulrich Hertrampf		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Grundkenntnisse in Mathematik und Theoretischer Informatik, wie sie in „Mathematik für Informatiker“ und „Theoretische Grundlagen der Informatik“ vermittelt werden.		
12. Lernziele:	Die Studierenden lernen die Grundprinzipien kennen, nach denen der im wesentlichen noch immer hypothetische Quantencomputer arbeiten würde. Sie kennen die grundlegenden Quantenalgorithmen von Deutsch, Jozsa, Simon und anderen, den Zusammenhang mit unitären Matrizen, wichtige Sätze wie das No-Cloning-Theorem. Sie haben den Quanten-Suchalgorithmus von Grover und den Primfaktorzerlegungs-Algorithmus von Shor verstanden.		
13. Inhalt:	Die Vorlesung behandelt zunächst die grundlegenden Techniken des Quanten-Computings: Qubits und Quantenregister, Messungen, Hadamard-Transformation, Quantenschaltkreise. Im zweiten Teil werden wichtige Algorithmen vorgestellt: Grovers Suchalgorithmus, Shors Faktorisierungsalgorithmus. Abschnitte über Teleportation und Quantenkryptographie runden das Thema ab.		
14. Literatur:	Matthias Homeister, „Quantum Computing verstehen“, 2. Auflage, Friedr. Vieweg & Sohn, 2008; Jozef Gruska, „Quantum computing“, McGraw-Hill, 1999.		
15. Lehrveranstaltungen und -formen:	517401 Vorlesung mit Übungen Quantencomputing		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	51741 Quantencomputing (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Formale Methoden der Informatik		

## Modul: 29670 Rapid Prototyping

2. Modulkürzel:	051230135	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Erfahrungen in mindestens einer Programmiersprache.		
12. Lernziele:	Die Studierenden beherrschen die schnelle Realisierung von Computing-Systemenausgehend von einer Algorithmen-Implementierung unter Verwendung eines Computer-Algebrasystems.		
13. Inhalt:	Text		
14. Literatur:	<ul style="list-style-type: none"> <li>• James O. Hamblen und Michael D. Furman, Rapid Prototyping of Digital Systems: A Tutorial Approach, 2001</li> <li>• More literature is named in the lecture</li> </ul>		
15. Lehrveranstaltungen und -formen:	296701 Vorlesung mit Übung Rapid Prototyping		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	29671 Rapid Prototyping (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

## Modul: 29680 Real-Time Programming

2. Modulkürzel:	051510301	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Erhard Plödereder		
9. Dozenten:	Erhard Plödereder		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Significant programming experience (not necessarily in real-time application) is highly advisable.</li> <li>• Knowledge of Ada, C/C++ and Unix is helpful, but not required.</li> </ul>		
12. Lernziele:	Students understand the standard terminology of deadline-driven, safety-critical real-time systems. They understand the issues that differentiate such systems from general software systems, and they know about available solutions, if any.		
13. Inhalt:	<ol style="list-style-type: none"> <li>1) General requirements and terminology of real-time systems</li> <li>2) Deterministic execution: avoiding language-, implementation- and hardware-induced non-determinisms; coping with limited resources; storage estimation and management; execution time estimation</li> <li>3) Fault tolerance: Faults and failure modes, N-version programming, voting, forward and backward recovery</li> <li>4) Simple scheduling regimes: cyclic executives, deadline guarantees</li> <li>5) Parallelism and priority scheduling regimes: processes, threads, tasks; run-time kernels; task management; interrupt handling</li> <li>6) Synchronization and communication: semaphores, critical regions, monitors, protected objects, rendezvous, messaging</li> <li>7) Control of shared resources</li> <li>8) Distributed Systems: basic concepts; major issues</li> </ol>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Alan Burns and Andy Wellings: Real-Time Systems and Programming Languages, Addison Wesley, 1997 ... or later editions of the Burns/Wellings-Book, e.g., 4.ed. 2009</li> <li>• Language reference manuals (C++, Java, Ada) are useful at times.</li> </ul>		
15. Lehrveranstaltungen und -formen:	296801 Vorlesung mit Übung Real-Time Programming		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	29681 Real-Time Programming (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Softwaretechnologie		

## Modul: 29690 Real-Time Video Processing I

2. Modulkürzel:	051230140	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	This course requires knowledge and experience in (at least) one programming language as well as knowledge of the subject "Technische Informatik or a similar course		
12. Lernziele:	The Students will gain knowledge in the implementation of algorithms, architectures and exemplary processors for real-time video processing		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Introduction: analog/digital Television</li> <li>• Cameras, Image sensors and their characteristics</li> <li>• Image Filtering, Bayer Filter</li> <li>• Motion Analysis</li> <li>• video compression</li> <li>• video communication</li> <li>• video processing</li> <li>• Parallel architecture, video processors and Implementation of hardware components for real-time video processing algorithms</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Roger Clarke und R. J. Clarke von Academic Press Inc, Digital Compression of Still Images and Video (Signal Processing and Its Applications), 1995</li> <li>• More literature is named in the lecture</li> </ul>		
15. Lehrveranstaltungen und -formen:	296901 Vorlesung mit Übung Real-Time Video Processing I		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	29691 Real-Time Video Processing I (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

## Modul: 29700 Real-Time Video Processing II

2. Modulkürzel:	051230142	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Voraussetzung für Real-Time Video Processing II sind Kenntnisse von Real-Time Video Processing I.Alternativ sind Kenntnisse aus einem Fach der Technischen Informatik oder einem ähnlichen Fach oder Kenntnisse im Bereich der Datenkompression oder der Bildverarbeitung oder der Signalverarbeitung Voraussetzung.		
12. Lernziele:	Die Studierenden können praktisch Real-Time Video Processing Systeme aufbauen.		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Vorstellung der Fallstudie eines Video Processing Systems</li> <li>• Auswahl der Algorithmen des Video Processing Systems</li> <li>• Implementierung und Verifikation der Algorithmen</li> <li>• Architektur-Entwicklung des Video Processing Systems</li> <li>• Performance-Analyse der Achitektur</li> <li>• Implementierung und System-Verifikation</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Roger Clarke und R. J. Clarke von Academic Press Inc, Digital Compression of Still Images and Video (Signal Processing and Its Applications), 1995</li> <li>• More literature is named in the lecture</li> </ul>		
15. Lehrveranstaltungen und -formen:	297001 Vorlesung mit Übung Real-Time Video Processing II		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	29701 Real-Time Video Processing II (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

## Modul: 45740 Rechnernetze II

2. Modulkürzel:	051200168	5. Moduldauer:	2 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Kurt Rothermel		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Kurt Rothermel</li> <li>• Frank Dürr</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	Rechnernetze		
12. Lernziele:	<p>Das in der Veranstaltung Rechnernetze I erworbene Verständnis von Konzepten, Protokollen und Technologien von Rechnernetzen wird weiter vertieft. Der Teilnehmer kennt die Funktionsweise der wichtigsten Dienste der Anwendungsschicht des Schichtenmodells. Er kennt die Konzepte zur Realisierung von Netzen auf Anwendungsebene. Er ist im Stande, diese Dienste und Konzepte bei der Konzeption eigener Anwendungen zu nutzen und ebenso eigene Dienste und Systeme zu entwickeln, um konkrete Probleme zu lösen.</p>		
13. Inhalt:	<p>Vorlesung Höhere Kommunikationskonzepte und -Protokolle:</p> <ol style="list-style-type: none"> <li>1. Einführung</li> <li>2. Socket-Schnittstelle</li> <li>3. Präsentation und Kompression</li> <li>4. Realzeitkommunikation</li> <li>5. Elektronische Bezahlssysteme</li> <li>6. Multicast auf Anwendungsschicht</li> <li>7. Inhaltsbezogene Netze</li> <li>8. Geographische Kommunikation</li> </ol> <p>Vorlesung Peer-to-Peer-Systeme:</p> <ol style="list-style-type: none"> <li>1. Definition und Anwendungen von Peer-to-Peer-Systemen</li> <li>2. Theoretische Netzmodelle</li> <li>3. Unstrukturierte Peer-to-Peer-Systeme</li> <li>4. Strukturierte Peer-to-Peer-Systeme</li> <li>5. Komplexe Suchanfragen</li> <li>6. Sicherheitsmechanismen für Peer-to-Peer-Systeme</li> <li>7. Software-Rahmenwerke für Peer-to-Peer-Systeme</li> </ol>		
14. Literatur:	<ul style="list-style-type: none"> <li>• J.F. Kurose, K.W. Ross, Computer Networking. 4th Edition, 2007</li> <li>• L.L. Peterson, B.S. Davie, Computer Networks: A Systems Approach. 4th Edition, 2007</li> <li>• Peter Mahlmann, Christian Schindelbauer, P2P Netzwerke: Algorithmen und Methoden., 2007</li> </ul>		



## Modul: 48580 Reinforcement Learning

2. Modulkürzel:	051200888	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Marc Toussaint		
9. Dozenten:	Vien Ngo		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	Solid knowledge in linear algebra, probability theory and optimization. Rough knowledge of Artificial Intelligence. Fluency in at least one programming language.		
12. Lernziele:	Students will acquire a deep understanding of Reinforcement Learning methods. Reinforcement Learning addresses the problem of learning optimal behavior (strongly related to optimal control) from data. This course will enable students to apply Reinforcement Learning algorithms in simulated domains and real robotic systems.		
13. Inhalt:	<p>Reinforcement Learning considers how an agent, interacting with a world, can improve or learn optimal behavior based on own experience or teacher demonstration. This branch of Artificial Intelligence and Machine Learning has become increasingly important as a foundation of robust intelligent systems and robotics. Optimal exploration (behavior that optimizes the agent's information gain) is a particularly interesting aspect of Reinforcement Learning. This lecture will introduce to the theory of Reinforcement Learning and then discuss state-of-the-art algorithms in this area.</p> <p>motivation and history          Markov Decision Processes and Bellman's optimality principle          relations to stochastic optimal control theory          basic model-free RL methods (TD-Learning, Q-learning, etc)          model-based RL methods          theory of optimal exploration (Bayesian RL, R-max)          relational RL          inverse RL, learning from demonstration and instruction          information theoretic formulations of RL          modern policy search methods (and applications in robotics)</p>		
14. Literatur:	<ul style="list-style-type: none"> <li>- (Main background) R. Sutton and A. Barto, Reinforcement Learning, 1998. This book is freely available online.</li> <li>- (For robotics application) S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics, 2006.</li> <li>- (Hardcore theory) C. Szepesvari, Algorithms for Reinforcement Learning, 2010. Draft version is freely available online.</li> <li>- S. LaValle, Planning Algorithms, 2006. <a href="http://planning.cs.uiuc.edu/">http://planning.cs.uiuc.edu/</a></li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 485801 Lecture Reinforcement Learning</li> <li>• 485802 Exercise Reinforcement Learning</li> </ul>		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit: 42 Stunden          Selbststudium: 138 Stunden</p>		

17. Prüfungsnummer/n und -name: 48581 Reinforcement Learning (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0

---

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von: Institut für Parallele und Verteilte Systeme

---

## Modul: 48600 Robotics I

2. Modulkürzel:	051200999	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Marc Toussaint		
9. Dozenten:	Marc Toussaint		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Solid knowledge in linear algebra, probability theory and optimization. Fluency in at least one programming language.		
12. Lernziele:	Students will acquire the basic methodologies to model, control and navigate robots, including trajectory planning, control of dynamic systems and object manipulation.		
13. Inhalt:	The lecture will give an introduction to robotics, focusing on essential theoretical foundations of planning and controlling motion, state estimation and eventually object manipulation. Exercises in simulations and on a real robot are a core element of this lecture to gain practical experience. <ul style="list-style-type: none"> <li>• motivation and history</li> <li>• (inverse) kinematics</li> <li>• path finding and trajectory optimization</li> <li>• (non-)holonomic systems</li> <li>• mobile robots</li> <li>• sensor processing (vision, range sensors)</li> <li>• simulation of robots and environments</li> <li>• object grasping and manipulation</li> </ul>		
14. Literatur:			
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 486001 Lecture Robotics I</li> <li>• 486002 Exercise Robotics I</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	48601 Robotics I (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Parallele und Verteilte Systeme		

## Modul: 48620 Scientific Visualization

2. Modulkürzel:	051900777	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Thomas Ertl		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Thomas Ertl</li> <li>• Daniel Weiskopf</li> <li>• Steffen Frey</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	<p>Basic concepts of Human Computer Interaction          Basic concepts of Computer Graphics</p>		
12. Lernziele:	<p>Student gains expertise about fundamental concepts and techniques of scientific visualization. This includes algorithms and mathematical background, data structures and implementation aspects as well as practical experience with widely available visualization tools.</p>		
13. Inhalt:	<p>Visualization discusses all aspects of visual representations of data gained from experiments, simulations, medical scanning machines, data bases an the like. The aim of visualization is to gain further insights into the data or the generate "simple" representations of complex phenomena or issues. For that, known techniques from the research area of interactive computer graphics as well as novel techniques are applied.</p> <p>The following topics will be discussed:</p> <ul style="list-style-type: none"> <li>• Introduction, history, visualization pipeline</li> <li>• Data aquisition and representation (sampling, reconstruction, grids, data structures)</li> <li>• PerceptionBasic concepts of visual mappings</li> <li>• Visualization of scalar fields (extraction of iso-surfaces, volume rendering)</li> <li>• Visualization of vector fields (particle tracking, texture-based methods, topology)</li> <li>• Tensor fields, multivariate data</li> <li>• Highdimensional data and information visualization</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• C. D. Hansen, C. R. Johnson, The Visualization Handbook, 2005</li> <li>• C. Ware, Information Visualization: Perception for Design, 2004</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 486201 Lecture Scientific Visualization</li> <li>• 486202 Exercise Scientific Visualization</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 48621 Scientific Visualization (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0</li> </ul>		

- V Vorleistung (USL-V), schriftlich oder mündlich
- 

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von: Institut für Visualisierung und Interaktive Systeme

---

## Modul: 29510 Service Computing

2. Modulkürzel:	052010004	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Frank Leymann</li> <li>• Dimka Karastoyanova</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 5. Semester            → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT            →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester            → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:			
12. Lernziele:	<p>A diversity of technologies enable nowadays computer-based interactions on the Web and on the Internet. The aim of this course is to make the students familiar with some of the most pervasive technologies that come together to form the Web and the Internet as we know it, and that enable to build large-scale application systems.</p>		
13. Inhalt:	<p>At first, we will cover the Web-centric technologies that enable the interaction of humans with Web content, e.g. HTTP, SMTP, AJAX, CSS and MIME . On the server-side part of technology, we will treat several Java EE technologies such as portlets, servlets, and JSP.</p> <p>The second part of the course will cover a set of technologies that are prominent in the landscape of Service-Oriented Architecture (SOA). In a nutshell, SOA is a paradigm that advocates the creation of complex, value added applications by reusing and composing independent and loosely coupled (software) services. We will dissect prominent SOA concepts like service discovery, addressing, policies, Service Bus, coordination protocols and service compositions. The architectural concepts will be complemented with an outlook of the technologies that embody them in the landscape of enterprise computing. In particular, we will cover several XML-centric technologies that sit at the core of Web services, e.g. XSD, SOAP, WSDL and Policy. In addition to the SOAP-based approach to Web services, we will also explore their REST aspect. Building on this portfolio of technologies, we will discuss the relationships between Web service technologies and “hot” items on the enterprise computing agenda such as autonomic/organic computing and cloud computing.</p>		
14. Literatur:	<p>S. Weerawarana, F. Curbera, F. Leymann, T. Storey, D. Ferguson: “Web Services Platform Architecture”, Prentice Hall 2005</p> <p>G. Alonso, F. Casati, H. Kuno, V. Machiraju: “Web Services”, Springer 2004</p> <p>E. Wilde:“World Wide Web”, Springer 1999</p> <p>M.P. Papazoglou: “Web Services: Principles &amp; Technology”, Pearson Education Limited 2008</p>		

N.M. Josuttis: "SOA in Practice: The Art of Distributed System Design", O'Reilly 2007

Th. Erl: "SOA: Entwurfsprinzipien für serviceorientierte Architektur", Addison-Wesley 2008

D.A. Chappell: "Enterprise Service Bus", O'Reilley 2004

---

15. Lehrveranstaltungen und -formen:	295101 Vorlesung mit Übungen Service Computing
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 29511 Service Computing (PL), schriftlich oder mündlich, 60 Min., Gewichtung: 1.0, Wenn Prüfung in diesem Fach, dann K E I N E Prüfung in Services and Service Composition (auch nicht innerhalb der VTL) möglich! mündl. Prüfungsdauer: 30 Minuten</li> <li>• V Vorleistung (USL-V), schriftlich oder mündlich</li> </ul>
18. Grundlage für ... :	29520 Ausgewählte Themen des Service Computing
19. Medienform:	Lecture and accompanying exercises
20. Angeboten von:	Architektur von Anwendungssystemen

---

## Modul: 46660 Service Management and Cloud Computing, and Evaluation

2. Modulkürzel:	052000111	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	Kristof Klöckner		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt; Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	Service Computing, Business Process Management		
12. Lernziele:	The students will learn the basics of systems management and cloud computing.		
13. Inhalt:	<p>Cloud Computing is an emerging paradigm for consumption and delivery of IT based services, based on concepts derived from consumer internet services, like self-service, apparently unlimited or elastic resources and flexible sourcing options. In this course we will discuss the technical foundations of cloud computing, as well as the business models associated with it.</p> <p>We will start by looking at virtualization and service management as the technical underpinnings. We will then look at infrastructure services and platform services, with a particular focus on emerging programming models for the cloud. We will discuss the trade-offs made between consistency and availability as well as extensions to traditional programming models. We also look at the life-cycle of applications in the cloud.</p> <p>Finally, we will look some of the challenges of Software as a Service, like multi-tenancy.</p> <p>Throughout the course, we will look both at existing products and services as well as the theoretical underpinnings.</p> <p>The course will be held as a combination of lectures and participant discussion.</p>		
14. Literatur:	To be announced in the lecture.		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 466601 Vorlesung Service Management and Cloud Computing, and Evaluation</li> <li>• 466602 Excercise Service Management and Cloud Computing, and Evaluation</li> </ul>		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit: 42 Stunden</p> <p>Selbststudium: 138 Stunden</p>		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 46661 Service Management and Cloud Computing, and Evaluation (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0</li> <li>• V Vorleistung (USL-V), mündliche Prüfung, 30 Min.</li> </ul>		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Architektur von Anwendungssystemen		

## Modul: 42520 Services and Service Composition

2. Modulkürzel:	052010008	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Dimka Karastoyanova		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Vasilios Andrikopoulos</li> <li>• Dimka Karastoyanova</li> <li>• Frank Leymann</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:			
12. Lernziele:	<p>The students will learn the foundations of the SOA and REST Architectural styles and technologies that can be used for their realization. The concept of service and the principle of loose coupling will be clarified. The students will be able to realize Service based applications using the Web Service technology. The students will be knowledgeable of the concepts workflow, service composition and how to apply them using workflow languages in order to create complex, value-added applications.</p>		
13. Inhalt:	<p>Architectural styles: SOA and REST          Basic principles: loose coupling vs. tight coupling          Service Technologies (WSDL, Policy, WS-Addressing, SOAP)          Virtualization and Middleware (Service Bus,â€¦)          Basics of the Workflow Technology          Business Process Re-engineering          Workflow Life Cycle          Workflow Management System Architecture          Workflow Languages (FDL, BPEL)</p>		
14. Literatur:	<ul style="list-style-type: none"> <li>• F. Leymann, D. Roller, Production Workflow, 2000</li> <li>• S. Weerawarana, F. Curbera, F. Leymann, T. Storey, D. Ferguson, Web Services Platform Architecture, 2005</li> <li>• W. van der Aalst, K. van Hee, Workflow Management, 2002</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 425201 Vorlesung Services and Service Compositions</li> <li>• 425202 Übung Services and Service Compositions</li> </ul>		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit: 48 Stunden          Selbststudiumszeit: 132 Stunden</p>		
17. Prüfungsnummer/n und -name:	<p>42521 Services and Service Composition (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0, Wenn Prüfung in diesem Fach, dann <b>K E I N E</b> Prüfung in Service Computing und/oder Business Process Management (auch nicht innerhalb der VTL) möglich!</p>		
18. Grundlage für ... :			
19. Medienform:			

20. Angeboten von:

Institut für Architektur von Anwendungssystemen

---

## Modul: 56550 Software Verification

2. Modulkürzel:	[pord.modulcode]	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	André Hoorn		
9. Dozenten:	Dozenten der Informatik		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Theoretical CS and Software Engineering modules (eg. SEfST, SZS)		
12. Lernziele:	The course has the objective to provide theoretical and practical knowledge about the essential software verification techniques. The goal is to enable students to use and apply them successfully to real world code artifacts. In this respects students will understand the required concepts of the different software verification techniques. Of great importance are, beside standard and time verification techniques also probabilistic verification techniques - these will be explained in detail and students will be able to apply them for quality assurance of probabilistic properties like safety, reliability and performance.		
13. Inhalt:	This course will introduce the foundations of this area  Process algebraic specification software systems  Abstract interpretation and symbolic execution  Axiomatic semantics of software systems  Software model checking  Modeling time in computation and timed verification for software systems  Quantitative analysis and probabilistic model checking  Probabilistic symbolic execution  Statistical methods in software verification		
14. Literatur:	1) Michael Huth and Mark Ryan. Logic in Computer Science: Modelling and Reasoning about Systems, second edition. Cambridge University Press, 2004.  2) Christel Baier and Joost-Pieter Katoen. Principles of Model Checking. The MIT Press, 2008		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 565501 Vorlesung Software Verification</li> <li>• 565502 Übung Software Verification</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden  Nachbearbeitungszeit: 138 Stunden		

17. Prüfungsnummer/n und -name:
- 56551 Software Verification (PL), schriftlich oder mündlich, 90 Min.,  
Gewichtung: 1.0
  - V Vorleistung (USL-V), schriftlich, eventuell mündlich
- 

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von:

---

## Modul: 42840 Software-Recht

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	3.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	2.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Hon.-Prof. Volker Haug		
9. Dozenten:	Volker Haug		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	keine		
12. Lernziele:	Die Studierenden gewinnen einen Überblick über die wichtigsten Felder des Softwarerechts (s.u., Inhalt) und sind dadurch in der Lage, rechtliche Problemstellungen früher zu erkennen und ihnen durch geeignete Maßnahmen vorzubeugen. Zugleich können Sie bei auftretenden Rechtsfragen eine erste Einordnung vornehmen.		
13. Inhalt:	Nach einer terminologischen Klärung des rechtlichen Software-Begriffs werden in einem ersten Block die wichtigsten Schutzrechte für Software überblicksartig und mit besonderen Bezügen zu Softwarefragen vorgestellt, insbesondere der Urheber- und Patentrechtsschutz sowie der Markenrechtsschutz. Der zweite Hauptteil der Vorlesung befasst sich mit dem Software-Vertragsrecht, wobei es hier zunächst um verschiedene Vertragstypen mit spezifischen Problemstellungen geht (Kauf, Leasing, Miete, u.a.), bevor das Leistungsstörungenrecht zu den verschiedenen denkbaren Mängeln bei Softwareprodukten und ihrer Pflege behandelt wird.		
14. Literatur:	Wird in der Vorlesung bekannt gegeben		
15. Lehrveranstaltungen und -formen:	428401 Vorlesung Software-Recht		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 28 Stunden, Nachbearbeitungszeit: 56 Stunden, Gesamt 84 Stunden		
17. Prüfungsnummer/n und -name:	42841 Software-Recht (USL), schriftlich, eventuell mündlich, 30 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:	<ul style="list-style-type: none"> <li>• Vortrag</li> <li>• PowerPoint-Folien</li> <li>• Tafelanschriften</li> </ul>		
20. Angeboten von:			

## Modul: 60140 Sprachbau mit Language Workbenches

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Stefan Wagner		
9. Dozenten:	Markus Völter		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Ergänzungsmodule --&gt;Wahlmodule aus Master SWT          →</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester          → Vorgezogene Master-Module</p>		
11. Empfohlene Voraussetzungen:	<p>Compilerbau</p> <p>Objektorientierte Programmierung</p>		
12. Lernziele:	<p><i>Die Studenten verstehen, warum und wie Modelle für den praktizierenden Softwareentwickler nützlich sind: Kommunikation, Analyse, Synthese. Sie verstehen warum es in vielen Fällen sinnvoll ist, eigene (domänenspezifische) Sprachen zu bauen. Sie verstehen die Mechanismen um Sprachen zu bauen, insbesondere die Fähigkeiten moderner Language Workbenches. Die Studenten sind in der Lage mit JetBrains MPS Sprachen zu bauen.</i></p>		
13. Inhalt:	<p><i>Modellierung, Grammatiken, Projizierende Editoren, Typsysteme, Codegenerierung, Interpreter. Grundlagen des Sprachdesigns: Ausdrucksfähigkeit vs. Komplexität, Vollständigkeit, Modularisierung, verschiedene Notationen. Wichtige Sprachparadigmas, die man in DSLs wiederverwenden kann: imperativ, funktional, zustandsbasiert. Grundlagen der Arbeit mit MPS.</i></p> <p><i>Die Veranstaltung wird als Blockveranstaltung als Workshop ausgeführt, viele praktische Anteile. Die Klausur findet direkt am Ende der Blockveranstaltung statt.</i></p> <p><i>Der Zeitraum ist 22. Bis 26. August.</i></p> <p><i>Ort ist bei der itemis AG, Industriestrasse 6, Vaihingen (direkt neben dem Bhf)</i></p>		
14. Literatur:	Buch <a href="http://dslbook.org/">http://dslbook.org/</a> + ggfs. diverse wissenschaftliche Papiere		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 601401 Vorlesung Sprachbau</li> <li>• 601402 Übung Sprachbau</li> </ul>		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit in Stunden: 56</p> <p>Selbststudiumszeit in Stunden: 124</p>		
17. Prüfungsnummer/n und -name:	60141 Sprachbau mit Language Workbenches (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:	<i>Powerpoint, Tafel, Demos, Diskussionen, Selbstarbeit der Studenten</i>		

20. Angeboten von: Software-Engineering

---

## Modul: 29500 Visual Computing

2. Modulkürzel:	051900014	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Martin Fuchs		
9. Dozenten:	Martin Fuchs		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule -->Wahlmodule aus Master SWT → B.Sc. Softwaretechnik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Modul 051900002 Computergraphik		
12. Lernziele:	The students know theoretical foundations for visual computing and acquired practical expertise in its core techniques. They are able to acquire scenes with digital cameras, can model their behavior and create content for non-2D displays and camera-projector systems.		
13. Inhalt:	The class is concerned with the digital processing of visual information by means of computer vision, computer graphics and image processing. It covers the following three interlocking topic complexes: Image processing: <ul style="list-style-type: none"> <li>• mathematical basics of image representations</li> <li>• noise models and noise suppression (including morphological, bilateral, and non-local filters)</li> <li>• selected topics from discrete image processing on image regions (e.g. photo montage with graph cuts, texture synthesis and space-time video completion)</li> </ul> Measuring / displaying light: <ul style="list-style-type: none"> <li>• selected topics from simple optics (esp. thin lenses and their interactions with light)</li> <li>• geometric camera models and calibration, typical optical distortions and means to counter them</li> <li>• radiometric camera calibration and HDR imaging</li> <li>• measuring and displaying color</li> <li>• plenoptic imaging / integral photography techniques, light field rendering and light field displays</li> <li>• passive stereo</li> </ul> Combined camera / illumination systems <ul style="list-style-type: none"> <li>• camera - illumination systems and photometric stereo</li> <li>• active stereo and projector-camera systems</li> <li>• the light transport matrix, its measurement and applications</li> </ul> Throughout, the class equally covers both acquisition (camera) and displays systems.		
14. Literatur:	<ul style="list-style-type: none"> <li>• Andrew S. Glassner, Principles of Digital Image Synthesis, 1995</li> <li>• J. Foley, A. van Dam, S. Feiner, J. Hughes, Computer Graphics: Principle and Practice, 1990</li> <li>• Jähne, Bernd, Digitale Bildverarbeitung, 2005</li> <li>• Literatur, siehe Webseite zur Veranstaltung</li> <li>• M. Pharr, G. Humphreys, Physically Based Rendering, 2004</li> </ul>		
15. Lehrveranstaltungen und -formen:	295001 Vorlesung mit Übungen Visual Computing		



## 400 Schlüsselqualifikationen fachaffin

---

Zugeordnete Module: 16610 Studienprojekt-Pr

---

## Modul: 16610 Studienprojekt-Pr

2. Modulkürzel:	051520191	5. Moduldauer:	2 Semester
3. Leistungspunkte:	15.0 LP	6. Turnus:	jedes Semester
4. SWS:	10.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Stefan Wagner		
9. Dozenten:	Dozenten der Informatik		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009, 4. Semester → Schlüsselqualifikationen fachaffin B.Sc. Softwaretechnik, PO 2012, 4. Semester → Schlüsselqualifikationen fachaffin		
11. Empfohlene Voraussetzungen:	Die Module Einführung in die Softwaretechnik, Programmentwicklung und Software-Praktikum müssen vor Beginn des Studienprojekts absolviert sein.		
12. Lernziele:	Im Studienprojekt-Pr werden die Prinzipien der Kooperation in einem größeren, für die Praxis typischen Projekt angewendet und eingeübt. Dazu gehören die Kontakte zum Kunden (Anforderungsanalyse), die Projektplanung, die Kostenschätzung, die Qualitätssicherung und die Präsentation der Resultate in schriftlicher und mündlicher Form, auch die Techniken zur Konfliktlösung und zum Risiko-Management. Natürlich kommt auch das fachliche Wissen zur Realisierung eines Softwaresystems zum Zuge. Die Teilnehmer sind nach dem Projekt in der Lage, ein größeres Softwareprojekt zu organisieren und vollständig durchzuführen.		
13. Inhalt:	Die Teilnehmer entwickeln ein Softwaresystem nach Vorgaben des Kunden von der Angebotserstellung bis zur Übergabe. Störungen und Änderungen der Aufgabe im Projektverlauf sind normale Bestandteile des Projekts. Typisch beginnt das Studienprojekt mit der Erhebung der Anforderungen und der Anfertigung eines Angebots; darauf folgt die Entwicklung nach einem zu Beginn gewählten Prozessmodell. Das Projekt wird mit der Übergabe der Software in einer Präsentation abgeschlossen. Die Teilnehmer fertigen einen Bericht an, der die individuellen Leistungen erkennen lässt		
14. Literatur:			
15. Lehrveranstaltungen und -formen:	166101 Praktikum Studienprojekt-Pr		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	210 h	
	Selbststudiums- /	240 h	
	Nachbearbeitungszeit:		
	Summe:	450 h	
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 16611 Studienprojekt-Pr (LBP), schriftlich und mündlich, Gewichtung: 1.0, Ausreichende Leistungen im Studienprojekt-Pr werden in jedem der beiden Semester durch einen unbenoteten Schein bestätigt. Die Note im Studienprojekt-Pr wird auf der Grundlage der im Projekt gezeigten Leistungen und des Projektberichts, der die individuellen Beiträge der Teilnehmer angeben muss, bestimmt. Sie geht in die Gesamtnote des Studienprojekts mit dem Gewicht 5 ein; der andere Beitrag kommt von Studienprojekt-Th mit dem Gewicht 3.</li> <li>• V Vorleistung (USL-V), Sonstiges</li> </ul>		

- V Vorleistung (USL-V), Sonstiges
- 

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von:

Institut für Softwaretechnologie

---

## Modul: 81110 Bachelorarbeit Softwaretechnik

2. Modulkürzel:	050525002	5. Moduldauer:	1 Semester
3. Leistungspunkte:	12.0 LP	6. Turnus:	jedes Semester
4. SWS:	8.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Stefan Wagner		
9. Dozenten:	Dozenten der Informatik		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009 B.Sc. Softwaretechnik, PO 2012		
11. Empfohlene Voraussetzungen:	sämtliche Basis- und Kernmodule sowie mindestens 120 LP im Bachelorstudiengang Softwaretechnik		
12. Lernziele:	Die Bachelorarbeit soll zeigen, dass innerhalb einer vorgegebenen Frist eine Aufgabenstellung aus dem Bereich Softwaretechnik selbständig nach wissenschaftlichen Methoden bearbeitet und die Ergebnisse sachgerecht dargestellt werden können.		
13. Inhalt:	wird von Prüfer festgelegt		
14. Literatur:	wird vom Prüfer bekanntgegeben		
15. Lehrveranstaltungen und -formen:			
16. Abschätzung Arbeitsaufwand:	Summe: 360 LP		
17. Prüfungsnummer/n und -name:			
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			