



**Universität Stuttgart**

**Modulhandbuch**  
**Studiengang Bachelor of Science Softwaretechnik**  
**Prüfungsordnung: 2009**

Wintersemester 2015/16  
Stand: 06. Oktober 2015

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
Studiengangsmanager/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>5</b>
<b>Qualifikationsziele</b> .....	<b>6</b>
<b>100 Basismodule</b> .....	<b>7</b>
12060 Datenstrukturen und Algorithmen .....	8
17210 Einführung in die Softwaretechnik .....	10
10190 Mathematik für Informatiker und Softwaretechniker .....	12
10260 Programmierkurs .....	14
10280 Programmierung und Software-Entwicklung .....	16
16520 Software-Qualität .....	18
10940 Theoretische Grundlagen der Informatik .....	19
<b>200 Kernmodule</b> .....	<b>21</b>
11890 Algorithmen und Berechenbarkeit .....	22
14360 Einführung in die Technische Informatik .....	23
14370 Fachstudie Softwaretechnik .....	25
14390 Programmentwicklung .....	26
14480 Sichere und zuverlässige Softwaresysteme .....	27
16500 Software Engineering .....	28
16510 Software-Praktikum .....	29
<b>210 Kernmodul Studienprojekt</b> .....	<b>30</b>
16780 Studienprojekt-Th .....	31
<b>300 Ergänzungsmodule</b> .....	<b>32</b>
310 Katalog SWT .....	33
41590 Einführung in die Numerik und Stochastik für Softwaretechniker .....	34
10210 Mensch-Computer-Interaktion .....	36
10220 Modellierung .....	38
36530 Rechnerorganisation 1 .....	40
10330 Systemkonzepte und -programmierung .....	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
10090 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
10210 Mensch-Computer-Interaktion .....	62
10220 Modellierung .....	64
36530 Rechnerorganisation 1 .....	66
10330 Systemkonzepte und -programmierung .....	68
40090 Systemkonzepte und -programmierung .....	70
330 Katalog ISW .....	72
10030 Architektur von Anwendungssystemen .....	73
10060 Computergraphik .....	75

---

41590 Einführung in die Numerik und Stochastik für Softwaretechniker .....	77
10090 Grundlagen der Eingebetteten Systeme .....	79
10100 Grundlagen der Graphischen Ingenieursysteme .....	80
10110 Grundlagen der Künstlichen Intelligenz .....	82
56930 Grundlagen der Rechnerarchitektur .....	84
42410 Grundlagen des Wissenschaftlichen Rechnens .....	86
10170 Imaging Science .....	88
10180 Information Retrieval und Text Mining .....	90
10210 Mensch-Computer-Interaktion .....	91
10220 Modellierung .....	93
39040 Rechnernetze .....	95
36530 Rechnerorganisation 1 .....	97
10330 Systemkonzepte und -programmierung .....	99
40090 Systemkonzepte und -programmierung .....	101
<b>400 Schlüsselqualifikationen fachaffin .....</b>	<b>103</b>
16610 Studienprojekt-Pr .....	104
<b>81110 Bachelorarbeit Softwaretechnik .....</b>	<b>106</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  
                              10260 Programmierkurs  
                              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 051520005 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>Konkret:</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; sowohl „originär“ parallel, als auch parallelisierte Versionen bereits vorgestellter sequentieller Algorithmen</li> </ul>		
13. Inhalt:	<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>• Wahl der Datenstrukturen; Listen, Bäume, Graphen; deren Definitionen, deren Datenstrukturen</li> <li>• diverse interne und externe Such- und Sortierverfahren (z.B. Linear-, Binär-, Interpolationssuche, AVL-, B-Bäume, internes und externes Hashing, mehrere langsame Sortierungen, Heap-, Quick-, Bucket-, Mergesort)</li> <li>• diverse Graphenalgorithmen (DFS, BFS, Besuchssequenzen, topol. Traversierung, Zusammenhangskomponenten, minimale Spannbäume, Dijkstra-, Floyd- kürzeste Wege)</li> <li>• Algorithmen auf Mengen und Relationen (transitive Hüllen, Warshall)</li> <li>• Korrektheitsbegriff und -formalismen; Spezifikation und Implementierung</li> <li>• Einige parallele und parallelisierte Algorithmen</li> <li>• einfache Elemente paralleler Programmierung, soweit für obiges notwendig</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Appelrath H.J., Ludewig. J., Skriptum Informatik, 1999</li> </ul>		

---

	<ul style="list-style-type: none"><li>• Sedgewick, R., Algorithms in C, 1998</li></ul>								
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"><li>• 120601 Vorlesung Datenstrukturen und Algorithmen</li><li>• 120602 Übung Datenstrukturen und Algorithmen</li></ul>								
16. Abschätzung Arbeitsaufwand:	<table><tr><td>Präsenzzeit:</td><td>63 h</td></tr><tr><td>Selbststudiums- /</td><td>207</td></tr><tr><td>Nachbearbeitungszeit:</td><td></td></tr><tr><td>Summe:</td><td>270 h</td></tr></table>	Präsenzzeit:	63 h	Selbststudiums- /	207	Nachbearbeitungszeit:		Summe:	270 h
Präsenzzeit:	63 h								
Selbststudiums- /	207								
Nachbearbeitungszeit:									
Summe:	270 h								
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"><li>• 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.</li><li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich</li></ul>								
18. Grundlage für ... :									
19. Medienform:									
20. Angeboten von:									

---

## 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>• sowie entsprechende Programmiererfahrung</li> </ul>		
12. Lernziele:	<p>Die Veranstaltung liefert einen ersten Einblick in die Softwaretechnik. Sie ist abgestimmt auf die Software-Qualität im 1. und Programmentwicklung im 3. Semester.</p> <p>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</p>		
13. Inhalt:	<p>Die Vorlesung behandelt technische und andere Aspekte der Softwarebearbeitung, wie sie in der Praxis stattfindet. Die einzelnen Themen sind:</p> <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:	Apl. Prof. Wolfgang Rump		
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: 10260 Programmierkurs

2. Modulkürzel:	051520010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	3.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	2.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Univ.-Prof. Jonas Kuhn		
9. Dozenten:	Max Kisselew		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009, 2. Semester → Basismodule		
11. Empfohlene Voraussetzungen:			
12. Lernziele:	<p>Selbstständiges Erstellen von Programmen und Lösung von Programmieraufgaben in der Programmiersprache Python, mit einem Schwerpunkt auf Konzepten, die für die maschinelle Sprachverarbeitung und Computerlinguistik wichtig sind.</p> <p>--</p> <p>Independently writing programs and solving programming tasks in the programming language Python, with emphasis on concepts relevant for Natural Language Processing and Computational Linguistics.</p>		
13. Inhalt:	<p>Das Modul "Programmierkurs" richtet sich primär an Studierende der Maschinellen Sprachverarbeitung (3. Semester), Computerlinguistik und Digital Humanities. Vermittelt werden die wichtigsten Konzepte der Programmiersprache Python und praktische Erfahrung bei der Erstellung von Python-Programmen bei der Verarbeitung von sprachlichen Daten und Ressourcen.</p> <p>Die Modulveranstaltung und die Materialien sind in der Regel überwiegend englischsprachig; es werden jedoch deutschsprachige Hilfestellungen angeboten.</p> <p>--</p> <p>The module primarily targets students in Natural Language Processing (3rd semester), Computational Linguistics and Digital Humanities. It covers the key concepts of the programming language Python and provides practical experience in writing Python programs in the context of processing linguistic data and resources.</p> <p>Typically, the lectures of the module course as well as the materials are in English; however, students not fluent in English in the programming context will receive support in German.</p>		
14. Literatur:	<p>Folien.</p> <p>Slides.</p>		
15. Lehrveranstaltungen und -formen:	102601 Übung Programmierkurs		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit: 21 Stunden</p> <p>Nachbearbeitungszeit: 69 Stunden</p>		
17. Prüfungsnummer/n und -name:	10261 Programmierkurs (USL), Sonstiges, Gewichtung: 1.0, Übungsschein - Scheinkriterien werden zu Beginn der Modulveranstaltung angekündigt. Criteria for credit are announced at the beginning of the module course.		
18. Grundlage für ... :			
19. Medienform:			

20. Angeboten von: Institut für Maschinelle Sprachverarbeitung

---

## 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

---

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>
---------------------------------	---

---

18. Grundlage für ... :	12060 Datenstrukturen und Algorithmen
-------------------------	---------------------------------------

---

19. Medienform:	<ul style="list-style-type: none"><li>• Folien über Beamer</li><li>• Tafelanschrieb</li></ul>
-----------------	---

---

20. Angeboten von:	Software-Engineering
--------------------	----------------------

---

## 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

---

## 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:	<p>B.Sc. Softwaretechnik, PO 2009, 3. Semester → Kernmodule</p> <p>B.Sc. Softwaretechnik, PO 2012, 3. Semester → Kernmodule</p> <p>M.Sc. Softwaretechnik, PO 2012, 1. Semester → Auflagenmodule des Masters</p>		
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

---

17. Prüfungsnummer/n und -name:	14361 Einführung in die Technische Informatik (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0
---------------------------------	--

---

18. Grundlage für ... :	36530 Rechnerorganisation 1
-------------------------	-----------------------------

---

19. Medienform:	
-----------------	--

---

20. Angeboten von:	Institut für Rechnergestützte Ingenieursysteme
--------------------	--

---

## 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- /	138 h	
	Nachbearbeitungszeit:		
	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- /	138 h	
	Nachbearbeitungszeit:		
	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: 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:	Univ.-Prof. Stefan Wagner		
9. Dozenten:	Lars Grunske		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009, 4. Semester → Kernmodule B.Sc. Softwaretechnik, PO 2012, 4. Semester → Kernmodule M.Sc. Softwaretechnik, PO 2012, 2. Semester → Auflagenmodule des Masters		
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		

---

## 210 Kernmodul Studienprojekt

---

Zugeordnete Module: 16780 Studienprojekt-Th

---

## Modul: 16780 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 2009, 4. Semester → Kernmodul Studienprojekt		
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:	<ul style="list-style-type: none"> <li>• Deininger, Lichter, Ludewig, Schneider, Studien-Arbeiten, 5. Aufl. 2005</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 167801 Vorlesung Studienprojekt-Th</li> <li>• 167802 Seminar Studienprojekt-Th</li> </ul>		
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>• 16781 Studienprojekt-Th - Prüfung (PL), mündliche Prüfung, 25 Min., Gewichtung: 2.0</li> <li>• 16782 Studienprojekt-Th - Schein (LBP), 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

---

---

## 310 Katalog SWT

---

Zugeordnete Module:    10210 Mensch-Computer-Interaktion  
                              10220 Modellierung  
                              10330 Systemkonzepte und -programmierung  
                              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> </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> <li>• Modul 40090 Systemkonzepte und -programmierung</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: 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 style="width: 100%; border-collapse: collapse;"> <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>• 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: 10330 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>		
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> <p>Konzepte zur Synchronisation über gemeinsamen Speicher</p> <ul style="list-style-type: none"> <li>• Synchronisationsprobleme und -lösungen</li> <li>• Synchronisationswerkzeuge: Semaphor, Monitor</li> </ul> <p>Konzepte zur Kommunikation und Synchronisation mittels Nachrichtentransfer</p> <ul style="list-style-type: none"> <li>• Taxonomie: Kommunikation und Synchronisation</li> <li>• Nachrichten als Kommunikationskonzept</li> <li>• Höhere Kommunikationskonzepte</li> </ul> <p>Basisalgorithmen für Verteilte Systeme</p> <ul style="list-style-type: none"> <li>• Erkennung globaler Eigenschaften</li> <li>• Schnappschussproblem</li> </ul>		

---

	<ul style="list-style-type: none"><li>• Konsistenter globaler Zustand</li><li>• Verteilte Terminierung</li></ul> Praktische nebenläufige Programmierung in Java <ul style="list-style-type: none"><li>• Threads und Synchronisation</li><li>• Socketschnittstelle</li><li>• RMI Programmierung</li></ul>
14. Literatur:	<ul style="list-style-type: none"><li>• Literatur, siehe Webseite zur Veranstaltung</li></ul>
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"><li>• 103301 Vorlesung Systemkonzepte und -programmierung</li><li>• 103302 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>• 10331 Systemkonzepte und -programmierung (PL), schriftliche Prüfung, 90 Min., Gewichtung: 7.0</li><li>• 10332 Systemkonzepte und -programmierung - Übungsschein (LBP), schriftliche Prüfung, 30 Min., Gewichtung: 3.0</li></ul>
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	

---

## 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
	10090	Grundlagen der Eingebetteten Systeme
	10100	Grundlagen der Graphischen Ingenieursysteme
	10110	Grundlagen der Künstlichen Intelligenz
	10210	Mensch-Computer-Interaktion
	10220	Modellierung
	10330	Systemkonzepte und -programmierung
	36530	Rechnerorganisation 1
	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>• 29490 Services und Service Komposition</li><li>• 29510 Service Computing</li><li>• 29530 Business Process Management</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 051900001 Mensch-Computer-Interaktion</li> <li>• Modul 051240005 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

---

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>
---------------------------------	---

---

18. Grundlage für ... :	
-------------------------	--

---

19. Medienform:	
-----------------	--

---

20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme
--------------------	---

---

## 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: 10090 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 2009, 4. Semester → Ergänzungsmodule -->Katalog ISG → B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISW →		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 051700005 Rechnerorganisation</li> </ul>		
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:	<ul style="list-style-type: none"> <li>• J. Teich, Digitale Hardware/Software-Systeme, 2. Auflage, 2007</li> <li>• P. Marwedel, Embedded System Design, 2006</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 100901 Vorlesung Grundlagen der Eingebetteten Systeme</li> <li>• 100902 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>• 10091 Grundlagen der Eingebetteten Systeme (PL), schriftliche Prüfung, 90 Min., Gewichtung: 75.0</li> <li>• 10092 Grundlagen der Eingebetteten Systeme - Rechnerübungen (LBP), Sonstiges, Gewichtung: 25.0</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:	B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISG → B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISW → B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule -->Katalog ISG → B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule -->Katalog ISW →		
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:	Inhalte: <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:

---

## 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:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden  <b>Gesamt: 180 Stunden</b>		

- 
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>• Hans-Joachim Wunderlich</li> <li>• Rafal Baranowski</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

---

17. Prüfungsnummer/n und -name:	56931 Grundlagen der Rechnerarchitektur (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0
---------------------------------	---

---

18. Grundlage für ... :	
-------------------------	--

---

19. Medienform:	
-----------------	--

---

20. Angeboten von:	Rechnerarchitektur
--------------------	--------------------

---

## 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:	<p>080300100 Mathematik für Informatiker und Softwaretechniker und 051240005 Numerische und Stochastische Grundlagen der Informatik bzw. 051240006 Einführung in die Numerik und Stochastik für Softwaretechniker</p>		
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: 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> </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> <li>• Modul 40090 Systemkonzepte und -programmierung</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: 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: 10330 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>		
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> <p>Konzepte zur Synchronisation über gemeinsamen Speicher</p> <ul style="list-style-type: none"> <li>• Synchronisationsprobleme und -lösungen</li> <li>• Synchronisationswerkzeuge: Semaphore, Monitor</li> </ul> <p>Konzepte zur Kommunikation und Synchronisation mittels Nachrichtentransfer</p> <ul style="list-style-type: none"> <li>• Taxonomie: Kommunikation und Synchronisation</li> <li>• Nachrichten als Kommunikationskonzept</li> <li>• Höhere Kommunikationskonzepte</li> </ul> <p>Basisalgorithmen für Verteilte Systeme</p> <ul style="list-style-type: none"> <li>• Erkennung globaler Eigenschaften</li> <li>• Schnapsschussproblem</li> </ul>		

---

	<ul style="list-style-type: none"><li>• Konsistenter globaler Zustand</li><li>• Verteilte Terminierung</li></ul> Praktische nebenläufige Programmierung in Java <ul style="list-style-type: none"><li>• Threads und Synchronisation</li><li>• Socketschnittstelle</li><li>• RMI Programmierung</li></ul>
14. Literatur:	<ul style="list-style-type: none"><li>• Literatur, siehe Webseite zur Veranstaltung</li></ul>
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"><li>• 103301 Vorlesung Systemkonzepte und -programmierung</li><li>• 103302 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>• 10331 Systemkonzepte und -programmierung (PL), schriftliche Prüfung, 90 Min., Gewichtung: 7.0</li><li>• 10332 Systemkonzepte und -programmierung - Übungsschein (LBP), schriftliche Prüfung, 30 Min., Gewichtung: 3.0</li></ul>
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	

---

## 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:	<p>Präsenzzeit: 42 Stunden</p> <p>Nachbearbeitungszeit: 138 Stunden</p>
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
	10090	Grundlagen der Eingebetteten Systeme
	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
	10330	Systemkonzepte und -programmierung
	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>• 29490 Services und Service Komposition</li><li>• 29510 Service Computing</li><li>• 29530 Business Process Management</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 051900001 Mensch-Computer-Interaktion</li> <li>• Modul 051240005 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

---

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>
---------------------------------	---

---

18. Grundlage für ... :	
-------------------------	--

---

19. Medienform:	
-----------------	--

---

20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme
--------------------	---

---

## 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: 10090 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 2009, 4. Semester → Ergänzungsmodule -->Katalog ISG → B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISW →		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 051700005 Rechnerorganisation</li> </ul>		
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:	<ul style="list-style-type: none"> <li>• J. Teich, Digitale Hardware/Software-Systeme, 2. Auflage, 2007</li> <li>• P. Marwedel, Embedded System Design, 2006</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 100901 Vorlesung Grundlagen der Eingebetteten Systeme</li> <li>• 100902 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>• 10091 Grundlagen der Eingebetteten Systeme (PL), schriftliche Prüfung, 90 Min., Gewichtung: 75.0</li> <li>• 10092 Grundlagen der Eingebetteten Systeme - Rechnerübungen (LBP), Sonstiges, Gewichtung: 25.0</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:	B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISG → B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISW → B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule -->Katalog ISG → B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule -->Katalog ISW →		
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:	Inhalte: <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:

---

## 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:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden  <b>Gesamt: 180 Stunden</b>		

- 
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>• Hans-Joachim Wunderlich</li> <li>• Rafal Baranowski</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

---

17. Prüfungsnummer/n und -name:	56931 Grundlagen der Rechnerarchitektur (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0
---------------------------------	---

---

18. Grundlage für ... :	
-------------------------	--

---

19. Medienform:	
-----------------	--

---

20. Angeboten von:	Rechnerarchitektur
--------------------	--------------------

---

## 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:	<p>080300100 Mathematik für Informatiker und Softwaretechniker und 051240005 Numerische und Stochastische Grundlagen der Informatik bzw. 051240006 Einführung in die Numerik und Stochastik für Softwaretechniker</p>		
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:	B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISW → B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISW →		
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> <li>• Orthogonal transforms such as cosine transform and wavelets</li> <li>• Compression: Generic compression (RLE, entropy coding), methods specialized to domain of images (e.g. jpeg)</li> <li>• Video: file formats, compression (e.g. mpeg)</li> <li>• Image enhancement and restauration</li> <li>• Basics of segmentation</li> </ul>		

14. Literatur:	<ul style="list-style-type: none"> <li>• Bässmann, Henning; Kreyss, Jutta, Bildverarbeitung Ad Oculos, 2004</li> <li>• Forsyth, David and Ponce, Jean, Computer Vision. A Modern Approach.: A Modern Approach Computer Vision. A Modern Approach, 2003</li> <li>• Gonzalez, Rafael C.; Woods, Richard E.; Eddins, Steven L., Digital Image Processing, 2004</li> <li>• Bigun, J.: Vision with Direction, 2006</li> <li>• Klaus D. Tönnies, Grundlagen der Bildverarbeitung, 2005</li> <li>• L. G. Shapiro, G. C. Stockman, Computer Vision, 2001</li> </ul>								
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 101701 Vorlesung Imaging Science</li> <li>• 101702 Übung Imaging Science</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>• 10171 Imaging Science (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 ... :	<ul style="list-style-type: none"> <li>• 29430 Computer Vision</li> <li>• 55640 Correspondence Problems in Computer Vision</li> </ul>								
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:	Univ.-Prof. Sebastian Pado		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Sebastian Pado</li> <li>• Roman Klinger</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule -->Katalog ISW → B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule -->Katalog ISW →		
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> </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> <li>• Modul 40090 Systemkonzepte und -programmierung</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:	B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISW → B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISW →		
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> <li>• J. F. Kurose, K. W. Ross, Computer Networks: a top-down approach featuring the Internet, 2001</li> <li>• L.L. Peterson, B.S. Davie, Computer Networks: A Systems Approach, 1999</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 390401 VL Rechnernetze</li> </ul>		

---

• 390402 ÜB Rechnernetze

---

16. Abschätzung Arbeitsaufwand:

Präsenzzeit: 42 Stunden  
Nachbearbeitungszeit: 138 Stunden

---

17. Prüfungsnummer/n und -name:

- 39041 Rechnernetze (PL), schriftlich oder mündlich, 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: 10330 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>		
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> <p>Konzepte zur Synchronisation über gemeinsamen Speicher</p> <ul style="list-style-type: none"> <li>• Synchronisationsprobleme und -lösungen</li> <li>• Synchronisationswerkzeuge: Semaphore, Monitor</li> </ul> <p>Konzepte zur Kommunikation und Synchronisation mittels Nachrichtentransfer</p> <ul style="list-style-type: none"> <li>• Taxonomie: Kommunikation und Synchronisation</li> <li>• Nachrichten als Kommunikationskonzept</li> <li>• Höhere Kommunikationskonzepte</li> </ul> <p>Basisalgorithmen für Verteilte Systeme</p> <ul style="list-style-type: none"> <li>• Erkennung globaler Eigenschaften</li> <li>• Schnapsschussproblem</li> </ul>		

---

	<ul style="list-style-type: none"><li>• Konsistenter globaler Zustand</li><li>• Verteilte Terminierung</li></ul> Praktische nebenläufige Programmierung in Java <ul style="list-style-type: none"><li>• Threads und Synchronisation</li><li>• Socketschnittstelle</li><li>• RMI Programmierung</li></ul>
14. Literatur:	<ul style="list-style-type: none"><li>• Literatur, siehe Webseite zur Veranstaltung</li></ul>
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"><li>• 103301 Vorlesung Systemkonzepte und -programmierung</li><li>• 103302 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>• 10331 Systemkonzepte und -programmierung (PL), schriftliche Prüfung, 90 Min., Gewichtung: 7.0</li><li>• 10332 Systemkonzepte und -programmierung - Übungsschein (LBP), schriftliche Prüfung, 30 Min., Gewichtung: 3.0</li></ul>
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	

---

## 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:	<p>Präsenzzeit: 42 Stunden</p> <p>Nachbearbeitungszeit: 138 Stunden</p>
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

---

## 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:

---

## 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:	Nach Ankuendigung
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		
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 Betreuer / Prüfer festgelegt		
14. Literatur:	wird vom Betreuer / 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:			