



Universität Stuttgart

Modulhandbuch
Studiengang Bachelor of Science Softwaretechnik
Prüfungsordnung: 2012

Sommersemester 2013
Stand: 25. März 2013

Universität Stuttgart
Keplerstr. 7
70174 Stuttgart

Kontaktpersonen:

Studiendekan/in:	Univ.-Prof.Dr. Daniel Weiskopf Institut für Visualisierung und Interaktive Systeme Tel.: E-Mail: daniel.weiskopf@vis.uni-stuttgart.de
Studiengangsmanager/in:	Corinna Vehlow Visualisierungsinstitut der Universität Stuttgart Tel.: E-Mail: corinna.vehlow@visus.uni-stuttgart.de
Prüfungsausschussvorsitzende/r:	Prof.Dr. Jochen Ludewig Institut für Softwaretechnologie Tel.: 0711/ 685-88354 E-Mail: jochen.ludewig@iste.uni-stuttgart.de
Fachstudienberater/in:	Bernhard Schmitz Institut für Visualisierung und Interaktive Systeme Tel.: E-Mail: Bernhard.Schmitz@vis.uni-stuttgart.de

Inhaltsverzeichnis

Qualifikationsziele	5
100 Basismodule	6
12060 Datenstrukturen und Algorithmen	7
17210 Einführung in die Softwaretechnik	9
10190 Mathematik für Informatiker und Softwaretechniker	11
10280 Programmierung und Software-Entwicklung	13
16520 Software-Qualität	15
10940 Theoretische Grundlagen der Informatik	16
200 Kernmodule	18
11890 Algorithmen und Berechenbarkeit	19
14360 Einführung in die Technische Informatik	20
14370 Fachstudie Softwaretechnik	22
14390 Programmentwicklung	23
36100 Programmierparadigmen	24
42790 Seminar SWT	25
14480 Sichere und zuverlässige Softwaresysteme	26
16500 Software Engineering	27
16510 Software-Praktikum	28
41940 Studienprojekt-Th	29
300 Ergänzungsmodule	30
31400 English for Software Engineering	31
320 Katalog ISG	32
10140 Advanced Processor Architecture	33
10030 Architektur von Anwendungssystemen	35
10060 Computergraphik	37
39250 Distributed Systems I	39
41590 Einführung in die Numerik und Stochastik für Softwaretechniker	41
18560 Grundlagen der Eingebetteten Systeme	43
10100 Grundlagen der Graphischen Ingenieursysteme	44
10110 Grundlagen der Künstlichen Intelligenz	46
10150 Grundlagen des Compilerbaus und der Programmiersprachen	48
42410 Grundlagen des Wissenschaftlichen Rechnens	50
10210 Mensch-Computer-Interaktion	52
10220 Modellierung	54
36530 Rechnerorganisation 1	56
40090 Systemkonzepte und -programmierung	58
330 Katalog ISW	60
10140 Advanced Processor Architecture	61
42480 Ausgewählte Kapitel des Wissenschaftlichen Rechnens	63
10040 Bildsynthese	64
10060 Computergraphik	66
41590 Einführung in die Numerik und Stochastik für Softwaretechniker	68
18560 Grundlagen der Eingebetteten Systeme	70
10100 Grundlagen der Graphischen Ingenieursysteme	71
10110 Grundlagen der Künstlichen Intelligenz	73
10150 Grundlagen des Compilerbaus und der Programmiersprachen	75
42410 Grundlagen des Wissenschaftlichen Rechnens	77
14380 Hardware Verification and Quality Assessment	79
10170 Imaging Science	81

10180 Information Retrieval und Text Mining	83
10210 Mensch-Computer-Interaktion	84
10220 Modellierung	86
39040 Rechnernetze	88
36530 Rechnerorganisation 1	90
40090 Systemkonzepte und -programmierung	92
11330 Visualisierung	94
310 Katalog SWT	96
41590 Einführung in die Numerik und Stochastik für Softwaretechniker	97
10150 Grundlagen des Compilerbaus und der Programmiersprachen	99
10210 Mensch-Computer-Interaktion	101
10220 Modellierung	103
36530 Rechnerorganisation 1	105
40090 Systemkonzepte und -programmierung	107
340 Wahlmodule aus Master SWT	109
29520 Ausgewählte Themen des Service Computing	110
10040 Bildsynthese	112
29530 Business Process Management	114
29430 Computer Vision	116
55640 Correspondence Problems in Computer Vision	118
29440 Geometric Modeling and Computer Animation	120
55630 Information Visualization and Visual Analytics	122
29480 Loose Coupling and Message Based Applications	124
42520 Services and Service Composition	126
29490 Services und Service Komposition	127
46760 Theoretical and Methodological Foundations of Visual Computing	128
29500 Visual Computing	130
400 Schlüsselqualifikationen fachaffin	132
16610 Studienprojekt-Pr	133

Qualifikationsziele

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

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

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

100 Basismodule

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

Modul: 12060 Datenstrukturen und Algorithmen

2. Modulkürzel:	051510005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	9.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	6.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Prof.Dr. Thomas Ertl		
9. Dozenten:	Stefan Funke		
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"> • Modul 051520005 Programmierung und Software-Entwicklung 		
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"> • Kenntnis der Eigenschaften elementarer und häufig benötigter Algorithmen • Verständnis für die Auswirkungen theoretischer und tatsächlicher Komplexität • Erweiterung der Kompetenz im Entwurf und Verstehen von Algorithmen und der zugehörigen Datenstrukturen • Erste Begegnung mit nebenläufigen Algorithmen; sowohl „originär“ parallel, als auch parallelisierte Versionen bereits vorgestellter sequentieller Algorithmen 		
13. Inhalt:	<ul style="list-style-type: none"> • Vorgehensweise bei der Entwicklung und Implementierung von Algorithmen • Komplexität und Effizienz von Algorithmen, O-Notation • Wahl der Datenstrukturen; Listen, Bäume, Graphen; deren Definitionen, deren Datenstrukturen • 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) • diverse Graphenalgorithmen (DFS, BFS, Besuchssequenzen, topol. Traversierung, Zusammenhangskomponenten, minimale Spannbäume, Dijkstra-, Floyd- kürzeste Wege) • Algorithmen auf Mengen und Relationen (transitive Hüllen, Warshall) • Korrektheitsbegriff und -formalismen; Spezifikation und Implementierung • Einige parallele und parallelisierte Algorithmen • einfache Elemente paralleler Programmierung, soweit für obiges notwendig 		
14. Literatur:	<ul style="list-style-type: none"> • Appelrath H.J., Ludewig. J., Skriptum Informatik, 1999 • Sedgewick, R., Algorithms in C, 1998 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 120601 Vorlesung Datenstrukturen und Algorithmen 		

	<ul style="list-style-type: none">• 120602 Übung Datenstrukturen und Algorithmen
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 63 Stunden Nachbearbeitungszeit: 207 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none">• 12061 Datenstrukturen und Algorithmen (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein. Die genauen Details der Übungsleistungen und Ihrer Anrechnung werden zu Beginn der Lehrveranstaltung bekannt gegeben.• V Vorleistung (USL-V), schriftlich, eventuell mündlich, 30 Min.
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.Dr. 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"> • 051520005 Programmierung und Software-Entwicklung • 051510005 Datenstrukturen und Algorithmen sowie entsprechende Programmiererfahrung		
12. Lernziele:	<p>EST ist, wie der Name sagt, die allgemeine Einführung 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.</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"> • Abgrenzung und Motivation des Software Engineerings • Vorgehensmodelle • Software-Management • Software-Prüfung und Qualitätssicherung • Methoden, Sprachen und Werkzeuge für die einzelnen Phasen: Spezifikation, Grobentwurf, Feinentwurf, Implementierung, Test 		
14. Literatur:	<ul style="list-style-type: none"> • Ludewig, Licher: Software Engineering. dpunkt-Verlag, Heidelberg. 2. Aufl. 2010 • Pfleeger, Atlee: Software Engineering, Pearson. 2010 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 172101 Vorlesung Einführung in die Softwaretechnik • 172102 Übung Einführung in die Softwaretechnik 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 17211 Einführung in die Softwaretechnik (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0, Vorleistung: Schein; keine Hilfsmittel zugelassen. • V Vorleistung (USL-V), schriftlich, eventuell mündlich, 30 Min. 		
18. Grundlage für ... :	<ul style="list-style-type: none"> • 16500 Software Engineering • 16510 Software-Praktikum 		
19. Medienform:	<ul style="list-style-type: none"> • Folien am Beamer unterstützt durch Tafel und Overhead 		

-
- Dokumente, Links und Diskussionsforum in ILIAS

20. Angeboten von:

Institut für Softwaretechnologie

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.Dr. Wolfgang Rump		
9. Dozenten:	Wolfgang Rump		
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"> • Grundlagen (Aussagenlogik, Mengen, Relationen, Abbildungen, Zahlenmengen, Grundbegriffe der Algebra) • Lineare Algebra (Vektorräume, lineare Abbildungen, Matrizen, Determinanten, lineare Gleichungssysteme, Eigenwerte, Normalformen, Hauptachsentransformation, Skalarprodukte) • Analysis (Konvergenz, Zahlenfolgen und Zahlenreihen, stetige Abbildungen, Folgen und Reihen von Funktionen, spezielle Funktionen) 2. Semester: <ul style="list-style-type: none"> • Differential- und Integralrechnung (Funktionen einer und mehrerer Variablen, Ableitungen, Taylorentwicklungen, Extremwerte, Integration, Anwendungen) • Gewöhnliche Differentialgleichungen (elementar lösbare Differentialgleichungen, Existenz und Eindeutigkeit von Lösungen) 		
14. Literatur:	<ul style="list-style-type: none"> • Anna Sändig, Mathematik, Vorlesungskripte , SS 2007 • D. Hachenberger, Mathematik für Informatiker, 2005 • M. Brill, Mathematik für Informatiker, 2001 • P.Hartmann, Mathematik für Informatiker, 2002 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 101901 Vorlesung Mathematik 1 für Informatik und Softwaretechnik • 101902 Übung Mathematik 1 für Informatik und Softwaretechnik • 101903 Vorlesung Mathematik 2 für Informatik und Softwaretechnik • 101904 Übung Mathematik 2 für Informatik und Softwaretechnik 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 126 Stunden Nachbearbeitungszeit: 414 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 10191 Mathematik für Informatiker und Softwaretechniker (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Ein Übungsschein aus den beiden Veranstaltungen, jeweils im 1. oder 2. Fachsemester zu erwerben • V Vorleistung (USL-V), schriftlich, eventuell mündlich, 30 Min. 		

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

Modul: 10280 Programmierung und Software-Entwicklung

2. Modulkürzel:	051520005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	9.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	6.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof.Dr. Stefan Wagner		
9. Dozenten:	Bernhard Mitschang		
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 Mathematik Vorkurs wird empfohlen.		
12. Lernziele:	Die Teilnehmer 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"> • Die Programmiersprache Java und die virtuelle Maschine • Objekte, Klassen, Schnittstellen, Blöcke, Programmstrukturen, Kontrakte • Klassenmodellierung mit der UML • Objekterzeugung und -ausführung • Boolesche Logik • Verzweigungen, Schleifen, Routinen, Abstraktionen, Modularisierung, Variablen, Zuweisungen • Rechner, Hardware • Syntaxdarstellungen • Übersicht über Programmiersprachen und -werkzeuge • Grundlegende Datenstrukturen und Algorithmen • Vererbung, Polymorphe • Semantik • Programmierung graphischer Oberflächen • Übergang zum Software Engineering 		
14. Literatur:	<ul style="list-style-type: none"> • Appelrath, Hans-Jürgen und Ludewig, Jochen, "Skriptum Informatik - eine konventionelle Einführung", Verlag der Fachvereine Zürich und B.G. Teubner Stuttgart, 4. Auflage 1999 • Meyer, Bertrand, "Touch of Class", Springer-Verlag, 2009 • Savitch, Walter, "Java. An Introduction to Problem Solving and Programming", Pearson, 6. Auflage, 2012 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 102801 Vorlesung Programmierung und Softwareentwicklung • 102802 Übung Programmierung und Softwareentwicklung 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	63 Stunden	
	Vor-/Nachbearbeitungszeit:	187 Stunden	
	Prüfungsvorbereitung:	20 Stunden	

-
17. Prüfungsnummer/n und -name:
- 10281 Programmierung und Software-Entwicklung (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, Studienleistung: Übungsschein. Voraussetzungen werden zu Beginn vom Dozenten festgesetzt. Dazu gehören eine bestimmte Anzahl von Vorträgen in den Übungen und ein bestimmter Teil der Übungspunkte. Modulprüfung: schriftlich, 120 Minuten, keine Hilfsmittel
 - V Vorleistung (USL-V), schriftlich, eventuell mündlich, 30 Min.

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

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.Dr. Stefan Wagner		
9. Dozenten:	<ul style="list-style-type: none"> • Stefan Wagner • Ivan Bogicevic 		
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"> • gleichzeitiger Besuch der Programmierung und Softwareentwicklung 		
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"> • Ludwig J., Lichter, H., Software Engineering - Grundlagen, Menschen, Prozesse, Techniken, 2007 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 165201 Vorlesung Software-Qualität • 165202 Übung Software-Qualität 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 21 Stunden Nachbearbeitungszeit: 69 Stunden		
17. Prüfungsnummer/n und -name:	16521 Software-Qualität (USL), Studienbegleitend, Gewichtung: 1.0, Übungsschein; Scheinkriterien werden zu Beginn der Veranstaltung angekündigt.		
18. Grundlage für ... :	17210 Einführung in die Softwaretechnik		
19. Medienform:			
20. Angeboten von:	Institut für Softwaretechnologie		

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:	Prof.Dr. Volker Diekert		
9. Dozenten:	<ul style="list-style-type: none"> • Volker Diekert • Ulrich Hertrampf 		
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:	<ul style="list-style-type: none"> • Logik und Diskrete Strukturen: <p>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.</p> <ul style="list-style-type: none"> • Automaten und Formale Sprachen: <p>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.</p>		
13. Inhalt:	<ul style="list-style-type: none"> • Logik und Diskrete Strukturen: <p>Einführung in die Aussagenlogik; formale Sprache; Semantik (Wahrheitswerte); Syntax (Axiome und Schlussregeln); Normalformen; Hornformeln; aussagenlogische Resolution; Korrektheit und Vollständigkeit für die Aussagenlogik; Einführung in die Prädikatenlogik 1. Stufe; formale Sprache; Semantik und Syntax; Normalformen; Herbrand-Theorie; prädikatenlogische Resolution; Kombinatorik, Graphen, elementare Zahlentheorie: Rechnen mit Restklassen, endliche Körper, RSA-Verfahren.</p> <ul style="list-style-type: none"> • Automaten und Formale Sprachen: <p>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.</p>		
14. Literatur:	<ul style="list-style-type: none"> • John Hopcroft, Jeffrey Ullman, Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie, 1988 • Uwe Schöning, Theoretische Informatik - kurzgefasst, 1999 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 109401 Vorlesung Logik und Diskrete Strukturen • 109402 Übung Logik und Diskrete Strukturen • 109403 Vorlesung Automaten und Formale Sprachen 		

	• 109404 Übung Automaten und Formale Sprachen
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 84 Stunden Nachbearbeitungszeit: 276 Stunden
17. Prüfungsnummer/n und -name:	• 10941 Theoretische Grundlagen der Informatik (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein • V Vorleistung (USL-V), schriftlich, eventuell mündlich, 30 Min.
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Institut für Formale Methoden der Informatik

200 Kernmodule

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

Modul: 11890 Algorithmen und Berechenbarkeit

2. Modulkürzel:	050420020	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Prof.Dr. Stefan Funke		
9. Dozenten:	<ul style="list-style-type: none"> • Stefan Funke • Volker Diekert • Ulrich Hertrampf 		
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, 3. 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"> • John Hopcroft, Jeffrey Ullman, Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie, 1988 • Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Introduction to Algorithms (Second Edition), 2001 • Volker Diekert, Entwurf und Analyse effizienter Algorithmen (Vorlesungsskript), 2006 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 118901 Vorlesung Algorithmen und Berechenbarkeit • 118902 Übung Algorithmen und Berechenbarkeit 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiumszeit /	138 h	
	Nacharbeitszeit:		
	Gesamt:	180 h	
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 11891 Algorithmen und Berechenbarkeit (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein • V Vorleistung (USL-V), schriftlich, eventuell mündlich, 30 Min. 		
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.Dr. Otto Eggenberger		
9. Dozenten:	Otto Eggenberger		
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"> • Informationsdarstellung • Zahlendarstellung und Codes • Digitale Grundbausteine • Logische Funktionen, Speicherelemente • Befehlsausführung, Programmablauf Elektrotechnische Grundlagen <ul style="list-style-type: none"> • Physikalische Grundbegriffe • Elektrische Spannung, elektrischer Strom • Elektrische Netzwerke • Halbleiterbauelemente • Digitale Grundschaltungen Digitale Schaltungen <ul style="list-style-type: none"> • Schaltnetzwerke • Boolesche Algebra und Schaltalgebra • Darstellung und Minimierung von Schaltfunktionen • Rückkopplung, Zustandsbegriff • Automaten und sequentielle Netzwerke • Digitale Standardschaltungen • Entwurfsmethodik 		
14. Literatur:	Dirk W. Hoffman: Grundlagen der technischen Informatik, Hanser, 2007 Bernd Becker, Rolf Drechsler, Paul Molitor: Technische Informatik, Pearson Studium, 2005 Jörg Keller, Wolfgang J. Paul: Hardware Design, Teubner, 3. Aufl. 2005		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 143601 Vorlesung Einführung in die Technische Informatik • 143602 Gruppenübungen Einführung in die Technische Informatik 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 126 Stunden		
17. Prüfungsnummer/n und -name:	14361 Einführung in die Technische Informatik (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0		
18. Grundlage für ... :			

19. Medienform:

20. Angeboten von:

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.Dr. 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"> • Bestandene Prüfung "Software Engineering" 		
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"> • 143701 Praktikum Fachstudie Softwaretechnik • 143702 Teamarbeit an den beteiligten Instituten mit örtlicher fachlicher Betreuung 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 126 Stunden		
17. Prüfungsnummer/n und -name:	14371 Fachstudie Softwaretechnik (USL), Studienbegleitend, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

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.Dr. Stefan Wagner		
9. Dozenten:	<ul style="list-style-type: none"> • Stefan Wagner • Jan-Peter Ostberg 		
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"> • Programmierung und Softwareentwicklung • Einführung in die Softwaretechnik 		
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"> • Grundlagen der objektorientierten Programmierung • Spezifikation und Entwurf objektorientierter Programme mit UML • Vertiefte Programmierung in Java 		
14. Literatur:	<ul style="list-style-type: none"> • Rumbaugh, Jacobson, Booch, The unified modeling language reference manual, 2nd ed., 2004 • Rupp, Queins, Zengler, UML 2 glasklar: Praxiswissen für die UML-Modellierung, 3. Aufl. 2007 • Ullenboom: Java ist auch eine Insel. Galileo Computing, 8. Aufl. 2009 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 143901 Vorlesung Programmentwicklung • 143902 Übung Programmentwicklung 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 126 Stunden		
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"> • Folien am Beamer unterstützt durch Tafel und Overhead • Dokumente, Links und Diskussionsforen in ILIAS 		
20. Angeboten von:	Institut für Softwaretechnologie		

Modul: 36100 Programmierparadigmen

2. Modulkürzel:	051510010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Prof.Dr. Erhard Plödereder		
9. Dozenten:	Erhard Plödereder		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Kernmodule		
11. Empfohlene Voraussetzungen:	Programmiererfahrung in mindestens einer Programmier-sprache, vornehmlich Java, so wie z. B. im Modul "Programmierung und Software Entwicklung" (10280) erworben.		
12. Lernziele:	Die Studierenden haben grundlegende Konzepte von Programmiersprachen verstanden, die dem Erlernen weiterer Sprachen und dem vertieften Verständnis ihnen bekannter Sprachen dienlich sind. Sie haben deren Anwendung in mindestens einer weiteren Programmiersprache ihrer Ausbildung verstanden. Sie können ihre Kenntnisse in einfachen Programmen anwenden. Sie können weitere Programmiersprachen in ihrer akademischen und beruflichen Karriere schneller und präziser erlernen.		
13. Inhalt:	Überblick typischer Konzepte in Programmiersprachen und ihrer Auswirkungen auf die Sprache und deren Anwendung: Grundsätzliche Ausführungsmodelle, Speichermodelle und deren Konsequenzen, Datentypen und Typsysteme, unterschiedliche Bindungskonzepte und ihre Auswirkungen, objekt-orientierte Sprachkonzepte, Abstraktion und Kompositionsmechanismen, funktionale Sprachen. Eventuell werden auch Elemente der parallelen Programmierung und der Logik-Programmierung mit einbezogen. Die Vorlesung ist kein Streifzug durch diverse Programmiersprachen, sondern die Vorstellung zugrundeliegender Prinzipien, und ihrer Begründung aus der Sicht des Software Engineering, insbesondere der Zuverlässigkeit der Anwendung, und, wo nötig, der Implementierungsmodelle.		
14. Literatur:	<ul style="list-style-type: none"> • Sebesta, Robert, Concepts of Programming Languages, 2010 • weitere Literatur wird zu Beginn der Lehrveranstaltung und auf den Webseiten bekanntgegeben. 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 361001 Vorlesung Programmierparadigmen • 361002 Übung Programmierparadigmen 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 36101 Programmierparadigmen (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0 • V Vorleistung (USL-V), schriftlich, eventuell mündlich 		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 42790 Seminar SWT

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

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:	Univ.-Prof.Dr. Lars Grunske		
9. Dozenten:	Lars Grunske		
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"> Teilnahme an der Lehrveranstaltung "Einführung in die Softwaretechnik" 		
12. Lernziele:	Die Teilnehmer beherrschen Techniken, deren Anwendung die Zuverlässigkeit der Software verbessert oder garantiert. Dabei stehen formale Ansätze im Vordergrund.		
13. Inhalt:	<ul style="list-style-type: none"> Notationen und Verfahren zur formalen Beschreibung und Prüfung der Software 		
14. Literatur:	-		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> 144801 Vorlesung Sichere und zuverlässige Softwaresysteme 144802 Übung Sichere und zuverlässige Softwaresysteme 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	14481 Sichere und zuverlässige Softwaresysteme (PL), schriftlich und mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

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:	-
8. Modulverantwortlicher:	Univ.-Prof.Dr. 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		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • Einführung in die Softwaretechnik • Programmentwicklung 		
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"> • Geschichte des Software Engineerings • Organisationsaspekte der Software-Bearbeitung • Software-Prozesse, Prozess-Bewertung und -Verbesserung • Software-Wartung • Weitere ausgewählte Kapitel des Software Engineerings 		
14. Literatur:	<ul style="list-style-type: none"> • Ludewig J., Lichter, H., Software Engineering - Grundlagen, Menschen, Prozesse, Techniken, 2. Aufl. 2010 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 165001 Vorlesung Software Engineering • 165002 Übung Software Engineering 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	16501 Software Engineering (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0, Keine Hilfsmittel zugelassen.		
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:	-
8. Modulverantwortlicher:	Univ.-Prof.Dr. Stefan Wagner		
9. Dozenten:	<ul style="list-style-type: none"> • Stefan Wagner • Daniel Kulesz • Jan-Peter Ostberg • Ivan Bogicevic 		
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"> • Einführung in die Softwaretechnik • Gleichzeitiger Besuch der Programmentwicklung 		
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"> • Ludwig J., Lichter, H., Software Engineering - Grundlagen, Menschen, Prozesse, Techniken. dpunkt-Verlag, Heidelberg, 2. Aufl. 2010 		
15. Lehrveranstaltungen und -formen:	165101 Praktikum Software-Praktikum		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 126 Stunden		
17. Prüfungsnummer/n und -name:	16511 Software-Praktikum (USL), Studienbegleitend, Gewichtung: 1.0		
18. Grundlage für ... :	<ul style="list-style-type: none"> • 16610 Studienprojekt-Pr • 16780 Studienprojekt-Th 		
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:	Institut für Softwaretechnologie		

Modul: 41940 Studienprojekt-Th

2. Modulkürzel:	051520192	5. Moduldauer:	2 Semester
3. Leistungspunkte:	9.0 LP	6. Turnus:	unregelmäßig
4. SWS:	6.0	7. Sprache:	-
8. Modulverantwortlicher:	Univ.-Prof.Dr. Stefan Wagner		
9. Dozenten:	Dozenten der Informatik		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 4. Semester → Kernmodule		
11. Empfohlene Voraussetzungen:	Die Module Einführung in die Softwaretechnik, Programmentwicklung und Software-Praktikum müssen vor Beginn des Studienprojekts absolviert sein. Das Studienprojekt-Th bildet mit dem Studienprojekt-Pr eine Einheit; beide können nur zusammen begonnen werden. Die Vorleistungen (Scheine) aus dem Studienprojekt-Pr sind für die abschließende Prüfung des Studienprojekt-Th Voraussetzung.		
12. Lernziele:	Vorlesung und Seminar dienen dazu, theoretische Grundlagen zum Studienprojekt-Pr zu vermitteln und die Arbeit im Projekt zu reflektieren.		
13. Inhalt:	Die konkreten Themen und Inhalte hängen vom jeweiligen Projekt ab.		
14. Literatur:	Deiningner, Lichter, Ludewig, Schneider , Studien-Arbeiten, 5. Aufl. 2005		
15. Lehrveranstaltungen und -formen:			
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 63 Stunden Nachbearbeitungszeit: 189 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 41941 Studienprojekt-Th (PL), schriftlich oder mündlich, Gewichtung: 2.0 • 41942 Studienprojekt-Th - benotete Studienleistung (BSL), schriftlich oder mündlich, 25 Min., Gewichtung: 1.0 		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

300 Ergänzungsmodule

Zugeordnete Module:	310	Katalog SWT
	31400	English for Software Engineering
	320	Katalog ISG
	330	Katalog ISW
	340	Wahlmodule aus Master SWT

Modul: 31400 English for Software Engineering

2. Modulkürzel:	051900012	5. Moduldauer:	1 Semester
3. Leistungspunkte:	3.0 LP	6. Turnus:	jedes Semester
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	John Nixon		
9. Dozenten:			
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009, 6. Semester → Ergänzungsmodule B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule		
11. Empfohlene Voraussetzungen:			
12. Lernziele:			
13. Inhalt:			
14. Literatur:			
15. Lehrveranstaltungen und -formen:			
16. Abschätzung Arbeitsaufwand:			
17. Prüfungsnummer/n und -name:	31401 English for Software Engineering (USL), schriftlich, eventuell mündlich, 30 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

320 Katalog ISG

Zugeordnete Module:	10030	Architektur von Anwendungssystemen
	10060	Computergraphik
	10100	Grundlagen der Graphischen Ingenieursysteme
	10110	Grundlagen der Künstlichen Intelligenz
	10140	Advanced Processor Architecture
	10150	Grundlagen des Compilerbaus und der Programmiersprachen
	10210	Mensch-Computer-Interaktion
	10220	Modellierung
	18560	Grundlagen der Eingebetteten Systeme
	36530	Rechnerorganisation 1
	39250	Distributed Systems I
	40090	Systemkonzepte und -programmierung
	41590	Einführung in die Numerik und Stochastik für Softwaretechniker
	42410	Grundlagen des Wissenschaftlichen Rechnens

Modul: 10140 Advanced Processor Architecture

2. Modulkürzel:	051700010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Prof.Dr. Hans-Joachim Wunderlich		
9. Dozenten:	Hans-Joachim Wunderlich		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule → Katalog ISW</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • 051700005 Rechnerorganisation 		
12. Lernziele:	<p>Good understanding of the basic concepts used in modern CPUs and computing systems. Awareness of the challenges in modern processor design and the reasoning behind current and future design trends.</p>		
13. Inhalt:	<p>Classic topics in computer architecture as hardware/software interface are discussed as well as more advanced topics which include:</p> <ul style="list-style-type: none"> • Technology basics: Design patterns, fabrication, yield, test and reliability, cost and quality, scaling. • Performance: Frequency and instructions per clock cycle, performance analysis and optimization. • Power dissipation: Analysis and optimization of power and performance, power and scaling. • Computer arithmetic: Efficient hardware for basic arithmetic, implementation of exponential, logarithm and trigonometric functions, floating point arithmetic and standards, arithmetic pipelines and filter, real-world floating point implementations like the Cell SPE or SPARC. • Instruction parallelism: Super scalar computing, static and dynamic scheduling, out-of-order execution, VLIW-processors, multithreading • Parallel architectures: Shared memory and message passing, multi-core processors, multi-core systems on a chip and emerging many-core technologies found in current graphic accelerators • Memory hierarchy: Memory technology and cache design. • Fault tolerance for single processors and multi processor systems 		
14. Literatur:	<ul style="list-style-type: none"> • I. Koren, Computer Arithmetic Algorithms, 2001 		

- J. L. Hennessy, D. A. Patterson, Computer Architecture: A Quantitative Approach, 2012
- S. Iman, M. Pedram, Logic Synthesis for Low Power VLSI Designs, 1997

15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none">• 101401 Vorlesung Grundlagen der Rechnerarchitektur• 101402 Übung Grundlagen der Rechnerarchitektur
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10141 Advanced Processor Architecture (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	

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:		Prof.Dr. Frank Leymann	
9. Dozenten:		Frank Leymann	
10. Zuordnung zum Curriculum in diesem Studiengang:		B.Sc. Softwaretechnik, PO 2009, 6. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule → Katalog ISG	
11. Empfohlene Voraussetzungen:		Vorlesungen des Grundstudiums.	
12. Lernziele:		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.	
13. Inhalt:		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.	
14. Literatur:		<ul style="list-style-type: none"> • A. Silberschatz, H. F. Korth, S. Sudarshan, Database System Concepts, 2002 • B. Neubauer, T. Ritter, F. Stoinnski, CORBA Komponenten, 2004 • F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal, Pattern-orientierte Software Architektur - Ein Patternsystem, 1998 • F. Leymann, D. Roller, Production Workflow, 2000 • L. Hohmann, Beyond Software Architecture, 2003 • M. Fowler, Patterns of Enterprise Application Architecture, 2003 • P. Bernstein, E. Newcomer, Principles of Transaction Processing, 1997 • S. Conrad, W. Hasselbring, A. Koschel, R. Tritsch, Enterprise Application Integration, 2006 • S. Weerawarana, F. Curbera, F. Leymann, T. Storey, D. Ferguson, Web Services Platform Architecture, 2005 • W. Emmerich, Konstruktion von verteilten Objekten, 2003 	
15. Lehrveranstaltungen und -formen:		<ul style="list-style-type: none"> • 100301 Vorlesung Grundlagen der Architektur von Anwendungssystemen • 100302 Übung Grundlagen der Architektur von Anwendungssystemen 	

16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none">• 10031 Architektur von Anwendungssystemen (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0,• V Vorleistung (USL-V),
18. Grundlage für ... :	<ul style="list-style-type: none">• 29480 Loose Coupling and Message Based Applications• 29490 Services und Service Komposition• 29510 Service Computing• 29530 Business Process Management
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:	Prof.Dr. Thomas Ertl		
9. Dozenten:	<ul style="list-style-type: none"> • Thomas Ertl • Daniel Weiskopf • Martin Fuchs 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule → Katalog ISW</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • Modul 051900001 Mensch-Computer-Interaktion • Modul 051240005 Numerik und Stochastik. 		
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"> • Überblick über den Prozess der Bildsynthese • Graphische Geräte, visuelle Wahrnehmung, Farbsysteme • Grundlegende Rastergraphik und Bildverarbeitung • Raytracing und Beleuchtungsmodelle • 2D und 3D Geometrietransformationen, 3D Projektion • Graphikprogrammierung in OpenGL 3 • Texturen • Polygonale und hierarchische Modelle • Rasterisierung und Verdeckungsrechnung • Grundlagen der geometrischen Modellierung (Kurven, Flächen) • Räumliche Datenstrukturen <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"> • J. Encarnacao, W. Strasser, R. Klein, Graphische Datenverarbeitung (Band1 und 2), 1997 • J. Foley, A. van Dam, S. Feiner, J. Hughes, Computer Graphics: Principle and Practice, 1990 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 100601 Vorlesung Computergraphik • 100602 Übung Computergraphik 		

16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none">• 10061 Computergraphik (PL), schriftliche Prüfung, 60 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: 39250 Distributed Systems I

2. Modulkürzel:	051200015	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Prof.Dr. Kurt Rothermel		
9. Dozenten:	Kurt Rothermel		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule → Katalog ISG		
11. Empfohlene Voraussetzungen:	Programmierung und Software-Entwicklung Datenstrukturen und Algorithmen Systemkonzepte und -Programmierung		
12. Lernziele:	The Students will gain an understanding of the basic characteristics, concepts and methods of distributed systems. Furthermore, the ability to analyze existing distributed applications and platforms with regard to its specific properties will be obtained. The implementation of distributed applications as well as system platforms based on the shown methods of that course is another objective. Due to the knowledge provided in that course, the students will be able to communicate with other experts of other professional disciplines, about topics in the field of distributed systems.		
13. Inhalt:	<ol style="list-style-type: none"> 1. Introduction to distributed systems 2. System models 3. Communication: Messages, Remote Procedure Call (RPC), Remote Method Invocation RMI 4. Naming: Generating and Resolution 5. Time Management and clocks in distributed Systems: Applications, logical clocks, physical clocks, synchronization of clocks 6. Synchronization of processes : mutual exclusion 7. Global state: concepts, snapshot algorithms, distributed Debugging 8. Transaction management: Serializability, barrier methods, 2-phase-commit-protocols 9. Data replication: primary copy, consensus-protocols and other algorithms 10. Safety/Security: Methods for confidentiality, integrity, authentication and authorization 11. Broadcast-algorithms: processing model, broadcast-semantics and algorithms 		
14. Literatur:	Literatur, siehe Webseite zur Veranstaltung		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 392501 Vorlesung Verteilte Systeme • 392502 Übungen Verteilte Systeme 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 h Selbststudiumszeit / Nachbearbeitungszeit: 138 h Gesamt: 180 h		

-
17. Prüfungsnummer/n und -name:
- 39251 Distributed Systems I (PL), schriftliche Prüfung, 60 Min.,
Gewichtung: 1.0
 - V Vorleistung (USL-V), schriftlich, eventuell mündlich
-

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

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:	Dr. Dirk Pflüger		
9. Dozenten:	<ul style="list-style-type: none"> • Dirk Pflüger • Stefan Zimmer • Thomas Ertl • Daniel Weiskopf • Marc Alexander Schweitzer 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog SWT</p> <p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Katalog SWT</p>		
11. Empfohlene Voraussetzungen:	Mathematik für Informatiker und Softwaretechniker (Modulkürzel 080300100; 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"> • numerische Algorithmik • Gleitpunktzahlen und Gleitpunktarithmetik • Interpolation & Approximation • Integration • lineare Gleichungssysteme • Iterative Lösung linearer und nichtlinearer Gleichungen • gewöhnliche Differentialgleichungen • Stochastik 		

- Zufall und Unsicherheit
- diskrete und kontinuierliche Wahrscheinlichkeitsräume
- Asymptotik

14. Literatur:	<ul style="list-style-type: none">• Huckle, Schneider; Numerik für Informatiker• Schickinger T., Steger A.; Diskrete Strukturen, Band 2, 2002• Dahmen, Reusken; Numerik für Ingenieure
15. Lehrveranstaltungen und -formen:	
16. Abschätzung Arbeitsaufwand:	<ul style="list-style-type: none">• Präsenzzeit: 42 Stunden• Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	41591 Einführung in die Numerik und Stochastik für Softwaretechniker (PL), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Simulation großer Systeme

Modul: 18560 Grundlagen der Eingebetteten Systeme

2. Modulkürzel:	051711010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Prof.Dr.-Ing. Martin Radetzki		
9. Dozenten:	Martin Radetzki		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog ISW		
11. Empfohlene Voraussetzungen:	Modul 051700005 Rechnerorganisation		
12. Lernziele:	Methodische und werkzeunterstützte Konstruktion von Hardware und Software für auf spezifische Anwendungen hin optimierte eingebettete Systeme.		
13. Inhalt:	Einführung in Charakteristik und Anwendungsgebiete eingebetteter Systeme, hardwarenahe Softwareentwicklung, Software-Scheduling, eingebettete Algorithmen (digitale Signalverarbeitung, Kanal- und Quellencodierung am Beispiel Viterbi-Algorithmus und MPEG-Video-Codec), zustandsbasierte Modellierung (Statecharts), eingebettete Prozessoren (Microcontroller, digitale Signalprozessoren, ARM), Bussysteme und Speicher, Systemsynthese (Taskgraphen, Allokation, Bindung, Ablaufplanung/Scheduling)		
14. Literatur:	J. Teich, Digitale Hardware/Software-Systeme, 2. Auflage, 2007 P. Marwedel, Embedded System Design, 2006		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 100901 Vorlesung Grundlagen der Eingebetteten Systeme • 100902 Übung Grundlagen der Eingebetteten Systeme 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 18561 Grundlagen der Eingebetteten Systeme (PL), schriftlich, eventuell mündlich, Gewichtung: 75.0 • V Vorleistung (USL-V), schriftliche Prüfung 		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

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:	Prof.Dr. Dieter Roller		
9. Dozenten:	Dieter Roller		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009, 6. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, PO 2009, 6. 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"> • Kenntnis und Verständnis von Modellen bei der Produktentwicklung • Grundkenntnisse über die wichtigsten Modellarten, Algorithmen und Datenstrukturen und Techniken für den Datenaustausch 		
13. Inhalt:	Inhalte: <ul style="list-style-type: none"> • Anforderungen an CAD-Systeme • zweidimensionale Modelle • dreidimensionale Modelle • interaktive Modellerstellung • Einführung in die Modifikationstechnik u. parametrische Modellierung • Methoden zur Modellmodifikation • Grundlagen der parametrischen Modellierung • Ansätze und Verfahren zur parametrischen Variantenerzeugung • Ausgewählte Anwendungsbeispiele • Überblick über weitergehende Modellieransätze • Datenverwaltung in CAD 		
14. Literatur:	<ul style="list-style-type: none"> • D. Roller, CAD - Effiziente Anpassungs- und Variantenkonstruktion, Springer-Verlag • Literatur, siehe Webseite zur Veranstaltung 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 101001 Vorlesung Grundlagen der Graphischen Ingenieursysteme • 101002 Übung Grundlagen der Graphischen Ingenieursysteme 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 Stunden	
	Nachbearbeitungszeit:	138 Stunden	
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.Dr.-Ing. Andrés Bruhn		
9. Dozenten:			
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule → Katalog ISW B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule → Katalog ISW		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • Modul 080300100 Mathematik für Informatiker und Softwaretechniker 		
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"> • Intelligenz • Agentenbegriff • Problemlösen durch Suchen, Suchverfahren • Probleme mit Rand- und Nebenbedingungen • Spiele • Aussagen- und Prädikatenlogik • Logikbasierte Agenten, Wissensrepräsentation • Inferenz • Planen • Unsicherheit, probabilistisches Schließen • Probabilistisches Schließen über die Zeit • Sprachverarbeitung • Entscheidungstheorie 		
14. Literatur:	<ul style="list-style-type: none"> • S. Russell, P. Norvig, Künstliche Intelligenz, 2004 • G. F. Luger, Künstliche Intelligenz, 2001 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 101101 Vorlesung Grundlagen der Künstlichen Intelligenz • 101102 Übung Grundlagen der Künstlichen Intelligenz 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden Gesamt: 180 Stunden		

-
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: 10150 Grundlagen des Compilerbaus und der Programmiersprachen

2. Modulkürzel:	051510015	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:		Prof.Dr. Erhard Plödereder	
9. Dozenten:		Erhard Plödereder	
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog SWT</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog SWT</p>		
11. Empfohlene Voraussetzungen:	<p>Grundkenntnisse aus den Einführungsvorlesungen des Informatikgrundstudiums, sowie einige Erfahrungen mit Programmierung. Vorkenntnisse über formale Sprachen sind vorteilhaft, aber nicht zwingend.</p>		
12. Lernziele:	<p>Die Studierenden haben die Grundkenntnisse erlangt, die zur effizienten Verwendung von Lexer- und Parser-Generatoren zur Analyse von Eingabetexten nötig sind. Sie verstehen die grundlegende Funktionsweise mehrerer Parse-Verfahren und kennen deren grammatikalischen Einschränkungen. Sie haben gelernt, die Fehlermeldungen aus diesen Generatoren und den Compilern oder Interpretern richtig einzuordnen. Ferner haben sie durch Betrachtung der Implementierungsmodelle typischer Programmiersprachenkonstrukte Verständnis für das Ausführungsverhalten und für typische, gefährliche Fehlerquellen in Anwendungsprogrammen erlangt.</p>		
13. Inhalt:	<p>Compilerarchitekturen im Überblick; lexikalische und syntaktische Analyse von Texten mit formaler Grammatik, insb. von Programmiersprachen. Lexikalische Analyse: endliche Automaten und ihre Implementierung; Syntaxanalyse: diverse Parser-Strategien, ihre Implementierung und Eigenschaften. Methoden der automatischen Generierung von Analysatoren aus Spezifikationen der Grammatiken. Fehlererkennung und -behandlung. Analyse der statischen Semantik: Grundbegriffe und elementare Methoden. Eigenschaften von Programmiersprachen; Realisierung der Laufzeitsemantik prozeduraler Programmiersprachen aus Benutzersicht, insbesondere Implementierungsmodelle der Speicherverwaltung und</p>		

der Unterprogrammaufrufe. Vermeidung typischer Fehlerquellen und überraschender Probleme in Anwendungsprogrammen.

(Nach SS14 wird sich der programmiersprachliche Teil ändern.)

14. Literatur:	<ul style="list-style-type: none">• Aho, Sethi, Ullman, Compilers - Principles, Techniques, and Tools, 1988• Wilhelm, Maurer, Uebersetzerbau, 1997
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none">• 101501 Vorlesung Grundlagen des Compilerbaus und der Programmiersprachen• 101502 Übung Grundlagen des Compilerbaus und der Programmiersprachen
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10151 Grundlagen des Compilerbaus und der Programmiersprachen (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	

Modul: 42410 Grundlagen des Wissenschaftlichen Rechnens

2. Modulkürzel:	051240020	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Dr. Dirk Pflüger		
9. Dozenten:	<ul style="list-style-type: none"> • Dirk Pflüger • Stefan Zimmer • Marc Alexander Schweitzer 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → 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"> • Differentialgleichungen und mathematische Modellierung. • Approximation des Modells, Störungsanalyse, Filterung, Homogenisierung. • Diskretisierung: Finite Differenzen, Finite Elemente, Adaptivität. • Fehlerschätzer. • Schnelle Löser für lineare Gleichungssysteme. • Parallelisierung: Strategien und Lastbalancierung 		
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"> • 424101 Vorlesung Grundlagen des Wissenschaftlichen Rechnens • 424102 Übung Grundlagen des Wissenschaftlichen Rechnens 		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit: 42 Stunden Selbststudiumszeit: 138 Stunden</p>		
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:

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.Dr. Albrecht Schmidt		
9. Dozenten:	<ul style="list-style-type: none"> • Albrecht Schmidt • Thomas Ertl • Daniel Weiskopf 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog SWT</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog SWT</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • 051520005 Programmierung und Software-Entwicklung • 051200005 Systemkonzepte und -programmierung 		
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"> • Einführung in die Grundlagen der Mensch-Computer Interaktion, historische Entwicklung • Entwurfsprinzipien und Modelle für moderne Benutzungsschnittstellen und interaktive Systeme • Informationsverarbeitung des Menschen, Wahrnehmung, Motorik, Eigenschaften und Fähigkeiten des Benutzers 		

	<ul style="list-style-type: none"> • Interaktionskonzepte und -stile, Metaphern, Normen, Regeln und Style Guides • 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:	<ul style="list-style-type: none"> • 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:	<ul style="list-style-type: none"> • 102101 Vorlesung Mensch-Computer-Interaktion • 102102 Übung Mensch-Computer-Interaktion
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 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:	Prof.Dr. Frank Leymann		
9. Dozenten:	<ul style="list-style-type: none"> • Bernhard Mitschang • Frank Leymann 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog SWT</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog SWT</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • 051520005 Programmierung und Software-Entwicklung • 051510005 Datenstrukturen und Algorithmen • 051200005 Systemkonzepte und -programmierung 		
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"> • Entity-Relationship Modell & komplexe Objekte • Relationenmodell & Relationenalgebra , Überblick SQL • Transformationen von ER nach Relationen, Normalisierung • XML, DTD, XML-Schema, Info-Set, Namensräume • Metamodelle & Repository • RDF, RDF-S & Ontologien • UML • Petri Netze, Workflownetze • BPMN 		
14. Literatur:	<ul style="list-style-type: none"> • A. Silberschatz, H. F. Korth, S. Sudarshan, Database System Concepts, 2002 • R. Eckstein, S. Eckstein, "XML und Datenmodellierung", dpunkt.verlag 2004 • M. Hitz, G. Kappel, E. Kapsammer, W. Retschitzegger, UML @ Work - Objektorientierte Modellierung mit UML2, 2005 • P. Hitzler, M. Krötzsch, S. Rudolph, Y. Sure, Semantic Web, 2008 • T.J. Teorey, Database Modeling & Design, 2nd Edition, 1994 		

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

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:	-
8. Modulverantwortlicher:		Prof.Dr. Hans-Joachim Wunderlich	
9. Dozenten:		Hans-Joachim Wunderlich	
10. Zuordnung zum Curriculum in diesem Studiengang:		B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog ISW B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog SWT B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule → Katalog ISW B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule → Katalog SWT	
11. Empfohlene Voraussetzungen:		Einführung in die Technische Informatik (14360)	
12. Lernziele:		Kenntnisse über die Grundlagen der Organisation von Rechnern und deren maschinennahe Programmierung Grundzüge über die Beschreibung und den Entwurf von Hardwaresystemen	
13. Inhalt:		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: <ul style="list-style-type: none"> • Grundstrukturen: Stack-, Akkumulator- und Register-basierende Rechner • Informationsdarstellung in Hardware: Daten- und Befehlsformate, Fehlererkennung und -korrektur, Befehlssätze und Unterstützung von Hochsprachen. • MIPS als RISC-Bespiel und seine Assemblerprogrammierung • Grundelemente und Entwurf auf Register-Transfer-Ebene • Grundzüge einer Hardware-Beschreibungssprache • Operationswerke: Multiplikation, Division, Gleitkommaeinheiten • Steuerwerksentwurf und Mikroprogrammierung • Befehlszyklus und Unterbrechungen • Pipelining und statisches Scheduling • Speicherorganisation: Cachestrukturen und virtueller Speicher, Seitenverwaltung, Segmentierung, TLB, MMU und DMA • Leistungsbewertung: Maßzahlen und CPI, Benchmarking und einfache Anwendung von Warteschlangen 	

14. Literatur:	Literatur, siehe Webseite zur Veranstaltung
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none">• 167901 Vorlesung Rechnerorganisation 1• 167902 Praktische und theoretische Übungen Rechnerorganisation 1• 365302 Praktische und theoretische Übungen Rechnerorganisation 1
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none">• 36531 Rechnerorganisation 1 (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0• V Vorleistung (USL-V), schriftliche Prüfung
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:	Prof.Dr. Kurt Rothermel		
9. Dozenten:	<ul style="list-style-type: none"> • Kurt Rothermel • Frank Leymann 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog SWT</p> <p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Katalog SWT</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> * Modul 051520005 Programmierung und Software-Entwicklung * Modul 051510005 Datenstrukturen und Algorithmen 		
12. Lernziele:	<ul style="list-style-type: none"> * Verstehen grundlegender Architekturen und Organisationsformen von Software-Systemen * Verstehen systemnaher Konzepte und Mechanismen * Kann existierende Systemplattformen und Betriebssysteme hinsichtlich ihrer Eigenschaften analysieren und anwenden. * Kann systemnahe Software entwerfen und implementieren. * Kann nebenläufige Programme entwickeln * Kann mit Experten anderer Fachgebiete die Anwendung von Systemfunktionen abstimmen. 		
13. Inhalt:	<p>Grundlegende Systemstrukturen - und organisationen</p> <ul style="list-style-type: none"> • Multitaskingsystem • Multiprozessorsystem • Verteiltes System <p>Modellierung und Analyse nebenläufiger Programme</p> <ul style="list-style-type: none"> • Abstraktionen: Atomare Befehle, Prozesse, nebenläufiges Programm • Korrektheit- und Leitungskriterien <p>Betriebssystemkonzepte</p> <ul style="list-style-type: none"> • Organisation von Betriebssystemen • Prozesse und Threads • Eingabe/Ausgabe • Scheduling 		

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

330 Katalog ISW

Zugeordnete Module:	10040	Bildsynthese
	10060	Computergraphik
	10100	Grundlagen der Graphischen Ingenieursysteme
	10110	Grundlagen der Künstlichen Intelligenz
	10140	Advanced Processor Architecture
	10150	Grundlagen des Compilerbaus und der Programmiersprachen
	10170	Imaging Science
	10180	Information Retrieval und Text Mining
	10210	Mensch-Computer-Interaktion
	10220	Modellierung
	11330	Visualisierung
	14380	Hardware Verification and Quality Assessment
	18560	Grundlagen der Eingebetteten Systeme
	36530	Rechnerorganisation 1
	39040	Rechnernetze
	40090	Systemkonzepte und -programmierung
	41590	Einführung in die Numerik und Stochastik für Softwaretechniker
	42410	Grundlagen des Wissenschaftlichen Rechnens
	42480	Ausgewählte Kapitel des Wissenschaftlichen Rechnens

Modul: 10140 Advanced Processor Architecture

2. Modulkürzel:	051700010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Prof.Dr. Hans-Joachim Wunderlich		
9. Dozenten:	Hans-Joachim Wunderlich		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule → Katalog ISW</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • 051700005 Rechnerorganisation 		
12. Lernziele:	<p>Good understanding of the basic concepts used in modern CPUs and computing systems. Awareness of the challenges in modern processor design and the reasoning behind current and future design trends.</p>		
13. Inhalt:	<p>Classic topics in computer architecture as hardware/software interface are discussed as well as more advanced topics which include:</p> <ul style="list-style-type: none"> • Technology basics: Design patterns, fabrication, yield, test and reliability, cost and quality, scaling. • Performance: Frequency and instructions per clock cycle, performance analysis and optimization. • Power dissipation: Analysis and optimization of power and performance, power and scaling. • Computer arithmetic: Efficient hardware for basic arithmetic, implementation of exponential, logarithm and trigonometric functions, floating point arithmetic and standards, arithmetic pipelines and filter, real-world floating point implementations like the Cell SPE or SPARC. • Instruction parallelism: Super scalar computing, static and dynamic scheduling, out-of-order execution, VLIW-processors, multithreading • Parallel architectures: Shared memory and message passing, multi-core processors, multi-core systems on a chip and emerging many-core technologies found in current graphic accelerators • Memory hierarchy: Memory technology and cache design. • Fault tolerance for single processors and multi processor systems 		
14. Literatur:	<ul style="list-style-type: none"> • I. Koren, Computer Arithmetic Algorithms, 2001 		

- J. L. Hennessy, D. A. Patterson, Computer Architecture: A Quantitative Approach, 2012
- S. Iman, M. Pedram, Logic Synthesis for Low Power VLSI Designs, 1997

15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none">• 101401 Vorlesung Grundlagen der Rechnerarchitektur• 101402 Übung Grundlagen der Rechnerarchitektur
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10141 Advanced Processor Architecture (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	

Modul: 42480 Ausgewählte Kapitel des Wissenschaftlichen Rechnens

2. Modulkürzel:	051240030	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Dr. Dirk Pflüger		
9. Dozenten:	<ul style="list-style-type: none"> • Dirk Pflüger • Stefan Zimmer • Marc Alexander Schweitzer 		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog ISW B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Katalog ISW		
11. Empfohlene Voraussetzungen:	080300100 Mathematik für Informatiker und Softwaretechniker und 051240005 Numerische und Stochastische Grundlagen der Informatik bzw. 051240006 Einführung in die Numerik und Stochastik für Softwaretechniker 051240020 Grundlagen des wissenschaftlichen Rechnens		
12. Lernziele:	Die Teilnehmer kennen ausgewählte aktuelle Forschungsthemen des wissenschaftlichen Rechnens und können mit der zugehörigen Primärliteratur arbeiten.		
13. Inhalt:	Aktuelle weiterführende Themen des wissenschaftlichen Rechnens, z.B. adaptive Finite Elemente, Fehlerschätzer, hierarchische Basen und dünne Gitter, robuste Multilevellöser, p-Version und Spektralverfahren.		
14. Literatur:	Primärliteratur zu den behandelten Themen: <ul style="list-style-type: none"> • Bungartz/Griebel: Sparse Grids; Acta Numerica, Volume 13, p. 147-269 • Quarteroni/Valli: Numerical approximation of partial differential equations • Quarteroni: Numerical models for differential problems 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 424801 Vorlesung Ausgewählte Kapitel des Wissenschaftlichen Rechnens • 424802 Übung Ausgewählte Kapitel des Wissenschaftlichen Rechnens 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudiumszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	42481 Ausgewählte Kapitel des Wissenschaftlichen Rechnens (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 10040 Bildsynthese

2. Modulkürzel:	051900012	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Prof.Dr. Thomas Ertl		
9. Dozenten:	<ul style="list-style-type: none"> • Martin Fuchs • Thomas Ertl • Daniel Weiskopf 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 6. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule → Wahlmodule aus Master SWT</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module</p> <p>M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • Modul 051900002 Computergraphik 		
12. Lernziele:	<p>Die Studierenden besitzen Wissen über verschiedene Ansätze und Algorithmen der dreidimensionalen Computergraphik, physikalisch-basierte Verfahren wie Raytracing und Radiosity, die den Lichttransport und die Wechselwirkung mit Materie modellieren, und numerische Methoden wie Monte-Carlo-Integration und Finite-Elemente-Verfahren die es erlauben, die Rendering-Gleichung zu lösen. Darüber hinaus kennen sie interaktive Verfahren, die unter Ausnutzung programmierbarer Grafik-Hardware realistische Beleuchtungseffekte in Echtzeit approximieren können, sowie bildbasierte Ansätze, die ohne geometrische Daten realistische Darstellungen erzeugen. Bild-basierte Verfahren verzichten auf eine geometrische Repräsentation der Szene und erzeugen neue Ansichten aus anderen aufgenommenen Bildern.</p>		
13. Inhalt:	<p>In dieser Vorlesung werden die folgenden Themen behandelt:</p> <ul style="list-style-type: none"> • Grafik Hardware und APIs, OpenGL • Texturen, prozedurale Modelle • Schattenberechnungen • Szenengraphen, Culling, Level-of-Detail Verfahren • Physikalisch-basierte Beleuchtungsberechnung, Fotorealistische Bildsynthese • Lokale Beleuchtungsmodelle • Raytracing, Monte-Carlo Methoden • Radiosity • Bild-basiertes Rendering 		
14. Literatur:	<ul style="list-style-type: none"> • Andrew S. Glassner, Principles of Digital Image Synthesis, 1995 • D. Eberly, 3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics, 2000 		

- J. Foley, A. van Dam, S. Feiner, J. Hughes, Computer Graphics: Principle and Practice, 1990
- Literatur, siehe Webseite zur Veranstaltung
- P. Dutre, P. Bekaert, K. Bala, Advanced Global Illumination, 2003
- Tomas Akenine-Möller, Eric Haines, Real-Time Rendering, 2002
- Matt Pharr, Greg Humphreys, Physically Based Rendering: From Theory To Implementation, Morgan Kaufmann; Auflage: 2nd revised edition. (26. August 2010)
- Peter Shirley et al, Fundamentals of Computer Graphics, Third Edition, A.K. Peters, July 2009

15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 100401 Vorlesung Bildsynthese • 100402 Übung Bildsynthese
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 10041 Bildsynthese (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0 • V Vorleistung (USL-V), schriftlich, eventuell mündlich, Übungsschein.
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme

Modul: 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:	Prof.Dr. Thomas Ertl		
9. Dozenten:	<ul style="list-style-type: none"> • Thomas Ertl • Daniel Weiskopf • Martin Fuchs 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule → Katalog ISW</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • Modul 051900001 Mensch-Computer-Interaktion • Modul 051240005 Numerik und Stochastik. 		
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"> • Überblick über den Prozess der Bildsynthese • Graphische Geräte, visuelle Wahrnehmung, Farbsysteme • Grundlegende Rastergraphik und Bildverarbeitung • Raytracing und Beleuchtungsmodelle • 2D und 3D Geometrietransformationen, 3D Projektion • Graphikprogrammierung in OpenGL 3 • Texturen • Polygonale und hierarchische Modelle • Rasterisierung und Verdeckungsrechnung • Grundlagen der geometrischen Modellierung (Kurven, Flächen) • Räumliche Datenstrukturen <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"> • J. Encarnacao, W. Strasser, R. Klein, Graphische Datenverarbeitung (Band1 und 2), 1997 • J. Foley, A. van Dam, S. Feiner, J. Hughes, Computer Graphics: Principle and Practice, 1990 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 100601 Vorlesung Computergraphik • 100602 Übung Computergraphik 		

16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none">• 10061 Computergraphik (PL), schriftliche Prüfung, 60 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: 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:	Dr. Dirk Pflüger		
9. Dozenten:	<ul style="list-style-type: none"> • Dirk Pflüger • Stefan Zimmer • Thomas Ertl • Daniel Weiskopf • Marc Alexander Schweitzer 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog SWT</p> <p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Katalog SWT</p>		
11. Empfohlene Voraussetzungen:	Mathematik für Informatiker und Softwaretechniker (Modulkürzel 080300100; 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"> • numerische Algorithmik • Gleitpunktzahlen und Gleitpunktarithmetik • Interpolation & Approximation • Integration • lineare Gleichungssysteme • Iterative Lösung linearer und nichtlinearer Gleichungen • gewöhnliche Differentialgleichungen • Stochastik 		

- Zufall und Unsicherheit
- diskrete und kontinuierliche Wahrscheinlichkeitsräume
- Asymptotik

14. Literatur:	<ul style="list-style-type: none">• Huckle, Schneider; Numerik für Informatiker• Schickinger T., Steger A.; Diskrete Strukturen, Band 2, 2002• Dahmen, Reusken; Numerik für Ingenieure
15. Lehrveranstaltungen und -formen:	
16. Abschätzung Arbeitsaufwand:	<ul style="list-style-type: none">• Präsenzzeit: 42 Stunden• Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	41591 Einführung in die Numerik und Stochastik für Softwaretechniker (PL), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Simulation großer Systeme

Modul: 18560 Grundlagen der Eingebetteten Systeme

2. Modulkürzel:	051711010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Prof.Dr.-Ing. Martin Radetzki		
9. Dozenten:	Martin Radetzki		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog ISW		
11. Empfohlene Voraussetzungen:	Modul 051700005 Rechnerorganisation		
12. Lernziele:	Methodische und werkzeunterstützte Konstruktion von Hardware und Software für auf spezifische Anwendungen hin optimierte eingebettete Systeme.		
13. Inhalt:	Einführung in Charakteristik und Anwendungsgebiete eingebetteter Systeme, hardwarenahe Softwareentwicklung, Software-Scheduling, eingebettete Algorithmen (digitale Signalverarbeitung, Kanal- und Quellencodierung am Beispiel Viterbi-Algorithmus und MPEG-Video-Codec), zustandsbasierte Modellierung (Statecharts), eingebettete Prozessoren (Microcontroller, digitale Signalprozessoren, ARM), Bussysteme und Speicher, Systemsynthese (Taskgraphen, Allokation, Bindung, Ablaufplanung/Scheduling)		
14. Literatur:	J. Teich, Digitale Hardware/Software-Systeme, 2. Auflage, 2007 P. Marwedel, Embedded System Design, 2006		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 100901 Vorlesung Grundlagen der Eingebetteten Systeme • 100902 Übung Grundlagen der Eingebetteten Systeme 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 18561 Grundlagen der Eingebetteten Systeme (PL), schriftlich, eventuell mündlich, Gewichtung: 75.0 • V Vorleistung (USL-V), schriftliche Prüfung 		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

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:	Prof.Dr. Dieter Roller		
9. Dozenten:	Dieter Roller		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009, 6. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, PO 2009, 6. 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"> • Kenntnis und Verständnis von Modellen bei der Produktentwicklung • Grundkenntnisse über die wichtigsten Modellarten, Algorithmen und Datenstrukturen und Techniken für den Datenaustausch 		
13. Inhalt:	Inhalte: <ul style="list-style-type: none"> • Anforderungen an CAD-Systeme • zweidimensionale Modelle • dreidimensionale Modelle • interaktive Modellerstellung • Einführung in die Modifikationstechnik u. parametrische Modellierung • Methoden zur Modellmodifikation • Grundlagen der parametrischen Modellierung • Ansätze und Verfahren zur parametrischen Variantenerzeugung • Ausgewählte Anwendungsbeispiele • Überblick über weitergehende Modellieransätze • Datenverwaltung in CAD 		
14. Literatur:	<ul style="list-style-type: none"> • D. Roller, CAD - Effiziente Anpassungs- und Variantenkonstruktion, Springer-Verlag • Literatur, siehe Webseite zur Veranstaltung 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 101001 Vorlesung Grundlagen der Graphischen Ingenieursysteme • 101002 Übung Grundlagen der Graphischen Ingenieursysteme 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 Stunden	
	Nachbearbeitungszeit:	138 Stunden	
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.Dr.-Ing. Andrés Bruhn	
9. Dozenten:			
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule → Katalog ISW B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule → Katalog ISW		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • Modul 080300100 Mathematik für Informatiker und Softwaretechniker 		
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"> • Intelligenz • Agentenbegriff • Problemlösen durch Suchen, Suchverfahren • Probleme mit Rand- und Nebenbedingungen • Spiele • Aussagen- und Prädikatenlogik • Logikbasierte Agenten, Wissensrepräsentation • Inferenz • Planen • Unsicherheit, probabilistisches Schließen • Probabilistisches Schließen über die Zeit • Sprachverarbeitung • Entscheidungstheorie 		
14. Literatur:	<ul style="list-style-type: none"> • S. Russell, P. Norvig, Künstliche Intelligenz, 2004 • G. F. Luger, Künstliche Intelligenz, 2001 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 101101 Vorlesung Grundlagen der Künstlichen Intelligenz • 101102 Übung Grundlagen der Künstlichen Intelligenz 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden Gesamt: 180 Stunden		

-
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: 10150 Grundlagen des Compilerbaus und der Programmiersprachen

2. Modulkürzel:	051510015	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:		Prof.Dr. Erhard Plödereder	
9. Dozenten:		Erhard Plödereder	
10. Zuordnung zum Curriculum in diesem Studiengang:		<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog SWT</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog SWT</p>	
11. Empfohlene Voraussetzungen:		<p>Grundkenntnisse aus den Einführungsvorlesungen des Informatikgrundstudiums, sowie einige Erfahrungen mit Programmierung. Vorkenntnisse über formale Sprachen sind vorteilhaft, aber nicht zwingend.</p>	
12. Lernziele:		<p>Die Studierenden haben die Grundkenntnisse erlangt, die zur effizienten Verwendung von Lexer- und Parser-Generatoren zur Analyse von Eingabetexten nötig sind. Sie verstehen die grundlegende Funktionsweise mehrerer Parse-Verfahren und kennen deren grammatikalischen Einschränkungen. Sie haben gelernt, die Fehlermeldungen aus diesen Generatoren und den Compilern oder Interpretern richtig einzuordnen. Ferner haben sie durch Betrachtung der Implementierungsmodelle typischer Programmiersprachenkonstrukte Verständnis für das Ausführungsverhalten und für typische, gefährliche Fehlerquellen in Anwendungsprogrammen erlangt.</p>	
13. Inhalt:		<p>Compilerarchitekturen im Überblick; lexikalische und syntaktische Analyse von Texten mit formaler Grammatik, insb. von Programmiersprachen. Lexikalische Analyse: endliche Automaten und ihre Implementierung; Syntaxanalyse: diverse Parser-Strategien, ihre Implementierung und Eigenschaften. Methoden der automatischen Generierung von Analysatoren aus Spezifikationen der Grammatiken. Fehlererkennung und -behandlung. Analyse der statischen Semantik: Grundbegriffe und elementare Methoden. Eigenschaften von Programmiersprachen; Realisierung der Laufzeitsemantik prozeduraler Programmiersprachen aus Benutzersicht, insbesondere Implementierungsmodelle der Speicherverwaltung und</p>	

der Unterprogrammaufrufe. Vermeidung typischer Fehlerquellen und überraschender Probleme in Anwendungsprogrammen.

(Nach SS14 wird sich der programmiersprachliche Teil ändern.)

14. Literatur:	<ul style="list-style-type: none">• Aho, Sethi, Ullman, Compilers - Principles, Techniques, and Tools, 1988• Wilhelm, Maurer, Uebersetzerbau, 1997
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none">• 101501 Vorlesung Grundlagen des Compilerbaus und der Programmiersprachen• 101502 Übung Grundlagen des Compilerbaus und der Programmiersprachen
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10151 Grundlagen des Compilerbaus und der Programmiersprachen (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	

Modul: 42410 Grundlagen des Wissenschaftlichen Rechnens

2. Modulkürzel:	051240020	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Dr. Dirk Pflüger		
9. Dozenten:	<ul style="list-style-type: none"> • Dirk Pflüger • Stefan Zimmer • Marc Alexander Schweitzer 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → 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"> • Differentialgleichungen und mathematische Modellierung. • Approximation des Modells, Störungsanalyse, Filterung, Homogenisierung. • Diskretisierung: Finite Differenzen, Finite Elemente, Adaptivität. • Fehlerschätzer. • Schnelle Löser für lineare Gleichungssysteme. • Parallelisierung: Strategien und Lastbalancierung 		
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"> • 424101 Vorlesung Grundlagen des Wissenschaftlichen Rechnens • 424102 Übung Grundlagen des Wissenschaftlichen Rechnens 		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit: 42 Stunden Selbststudiumszeit: 138 Stunden</p>		
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:

Modul: 14380 Hardware Verification and Quality Assessment

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

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

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Institut für Technische Informatik

Modul: 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.Dr.-Ing. 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"> • Modul 080300100 Mathematik für Informatiker und Softwaretechniker 		
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"> • Grundlagen aus der Optik:Lochkamera, Linsengleichung • Bildaufnahme:Kameras, Objektive, Beleuchtung, Aufnahmeprozess • Bildrepräsentation:Diskretisierung, Farbräume • Elementare Bildbearbeitung:Punktoperationen (z.B. Kontrastverstärkung, Binarisierung) • Lineare und nichtlineare Filter:Faltung, morphologische Operatoren • Fouriertransformation, Bilddarstellung und -bearbeitung im Fourierraum, Abtasttheorem • Orthogonale Transformationen:Cosinus, Wavelets • Kompression:Generische Verfahren (RLE, Entropie), spezielle Bildverfahren (z.B. jpeg) • Video:Formate, Kompression (z.B. MPEG) • Bildverbesserung und Restauration • Elementare Segmentierungsverfahren <ul style="list-style-type: none"> • Fundamentals of optics such as pinhole camera and lens equation • Image acquisition: Cameras, lenses, illumination, acquisition process • Image representation: Discretization, color spaces • Basics of image processing, e.g. point operations such as contrast enhancement or binarization • Linear and nonlinear filtering such as convolution and morphological operations. • Fourier transform, image representation and processing in Fourier space, sampling theorem • Orthogonal transforms such as cosine transform and wavelets • Compression: Generic compression (RLE, entropy coding), methods specialized to domain of images (e.g. jpeg) • Video: file formats, compression (e.g. mpeg) • Image enhancement and restauration • Basics of segmentation 		

14. Literatur:	<ul style="list-style-type: none"> • Bässmann, Henning; Kreys, Jutta, Bildverarbeitung Ad Oculos, 2004 • Forsyth, David and Ponce, Jean, Computer Vision. A Modern Approach.: A Modern Approach Computer Vision. A Modern Approach, 2003 • Gonzalez, Rafael C.; Woods, Richard E.; Eddins, Steven L., Digital Image Processing, 2004 • Bigun, J.: Vision with Direction, 2006 • Klaus D. Tönnies, Grundlagen der Bildverarbeitung, 2005 • L. G. Shapiro, G. C. Stockman, Computer Vision, 2001
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 101701 Vorlesung Imaging Science • 101702 Übung Imaging Science
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden Gesamt: 180 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 10171 Imaging Science (PL), schriftlich, eventuell mündlich, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben. • V Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	<ul style="list-style-type: none"> • 29430 Computer Vision • 55640 Correspondence Problems in Computer Vision
19. Medienform:	
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme

Modul: 10180 Information Retrieval und Text Mining

2. Modulkürzel:	052401010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Prof.Ph.D. Hinrich Schütze		
9. Dozenten:	<ul style="list-style-type: none"> • Helmut Schmid • Hinrich Schütze 		
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:	052400009		
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"> • Textpräprozessierung • invertierte Indexe • IR-Modelle (z.B. Vektorraum-basiertes IR) • Linkanalyse • Clustering • Frage-Antwort-Systeme • Informationsextraktion • korpusbasierter Erwerb von lexikalischem und Weltwissen 		
14. Literatur:	<ul style="list-style-type: none"> • Chris Manning, Prabhakar Raghavan, Hinrich Schütze, Introduction to Information Retrieval, 2008 Cambridge University Press 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 101801 Vorlesung Information Retrieval and Text Mining • 101802 Übung Information Retrieval and Text Mining 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	10181 Information Retrieval und Text Mining (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

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.Dr. Albrecht Schmidt		
9. Dozenten:	<ul style="list-style-type: none"> • Albrecht Schmidt • Thomas Ertl • Daniel Weiskopf 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog SWT</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog SWT</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • 051520005 Programmierung und Software-Entwicklung • 051200005 Systemkonzepte und -programmierung 		
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"> • Einführung in die Grundlagen der Mensch-Computer Interaktion, historische Entwicklung • Entwurfsprinzipien und Modelle für moderne Benutzungsschnittstellen und interaktive Systeme • Informationsverarbeitung des Menschen, Wahrnehmung, Motorik, Eigenschaften und Fähigkeiten des Benutzers 		

	<ul style="list-style-type: none"> • Interaktionskonzepte und -stile, Metaphern, Normen, Regeln und Style Guides • 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:	<ul style="list-style-type: none"> • 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:	<ul style="list-style-type: none"> • 102101 Vorlesung Mensch-Computer-Interaktion • 102102 Übung Mensch-Computer-Interaktion
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 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:	Prof.Dr. Frank Leymann		
9. Dozenten:	<ul style="list-style-type: none"> • Bernhard Mitschang • Frank Leymann 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog SWT</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog SWT</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • 051520005 Programmierung und Software-Entwicklung • 051510005 Datenstrukturen und Algorithmen • 051200005 Systemkonzepte und -programmierung 		
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"> • Entity-Relationship Modell & komplexe Objekte • Relationenmodell & Relationenalgebra , Überblick SQL • Transformationen von ER nach Relationen, Normalisierung • XML, DTD, XML-Schema, Info-Set, Namensräume • Metamodelle & Repository • RDF, RDF-S & Ontologien • UML • Petri Netze, Workflownetze • BPMN 		
14. Literatur:	<ul style="list-style-type: none"> • A. Silberschatz, H. F. Korth, S. Sudarshan, Database System Concepts, 2002 • R. Eckstein, S. Eckstein, "XML und Datenmodellierung", dpunkt.verlag 2004 • M. Hitz, G. Kappel, E. Kapsammer, W. Retschitzegger, UML @ Work - Objektorientierte Modellierung mit UML2, 2005 • P. Hitzler, M. Krötzsch, S. Rudolph, Y. Sure, Semantic Web, 2008 • T.J. Teorey, Database Modeling & Design, 2nd Edition, 1994 		

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

15. Lehrveranstaltungen und -formen:

- 102201 Vorlesung Modellierung
- 102202 Übung Modellierung

16. Abschätzung Arbeitsaufwand:

Präsenzzeit: 42 Stunden
Nachbearbeitungszeit: 138 Stunden

17. Prüfungsnummer/n und -name:

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

18. Grundlage für ... :

- 10030 Architektur von Anwendungssystemen
- 10080 Database and Information Systems

19. Medienform:

20. Angeboten von:

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:	Prof.Dr. Kurt Rothermel		
9. Dozenten:	<ul style="list-style-type: none"> • Kurt Rothermel • Frank Dürr 		
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"> • 051520005 Programmierung und Software-Entwicklung • 051510005 Datenstrukturen und Algorithmen • Grundkenntnisse in Java 		
12. Lernziele:	<ul style="list-style-type: none"> • Versteht grundsätzliche Eigenschaften, Konzepte und Methoden von Rechnernetzen, insbesondere dem Internet. • Versteht Schichten und deren Zusammenwirken in einem Protokollstapel • Kann Rechnernetze aufbauen, verwalten und analysieren. • Kann Protokolle entwickeln und in Schichtenarchitektur einbetten. • Kann höhere Kommunikationsdienste zur Entwicklung von netzgestützten Systemen anwenden. • Kann sich mit Experten anderer Domänen über Methoden der Rechnernetze verständigen. 		
13. Inhalt:	<ul style="list-style-type: none"> • Einführung in die Rechnernetze, ISO Referenzmodell; • Bitübertragungsschicht: Übertragungsmedien, analoge und digitale Informationskodierung und -übertragung, Vermittlungsarten; • Sicherungsschicht: Betriebsarten, Fehlererkennung und -behandlung, Flusskontrolle; • Lokale Netze: CSMA/CD, Token Ring, Token Bus, FDDI, Kopplung; • Vermittlungsschicht: Verbindungsorientierter und verbindungsloser Dienst, Leitwegbestimmung, Überlastkontrolle; • Internetworking; • Internet-Protokoll; • Transportschicht: ausgewählte Realisierungsprobleme und Internet-Protokolle; • Echtzeitkommunikation: IntServ, DiffServ; Sicherheit: Verfahren, IPsec, SSL, TLS. 		
14. Literatur:	<ul style="list-style-type: none"> • A.S. Tanenbaum, Computer Networks, 4th Edition, 2003 • D.E. Comer, Computernetzwerke und Internets, 2000 • D.E. Comer, Internetworking with TCP/IP Volume I: Principles, Protocols, and Architecture, 1995 • J. F. Kurose, K. W. Ross, Computer Networks: a top-down approach featuring the Internet, 2001 • L.L. Peterson, B.S. Davie, Computer Networks: A Systems Approach, 1999 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 390401 VL Rechnernetze 		

-
- 390402 ÜB Rechnernetze
-

16. Abschätzung Arbeitsaufwand:

Präsenzzeit: 42 Stunden
Nachbearbeitungszeit: 138 Stunden

17. Prüfungsnummer/n und -name:

- 39041 Rechnernetze (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0
 - V Vorleistung (USL-V), schriftlich, eventuell mündlich
-

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

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:	-
8. Modulverantwortlicher:	Prof.Dr. Hans-Joachim Wunderlich		
9. Dozenten:	Hans-Joachim Wunderlich		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog ISW B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog SWT B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule → Katalog ISW B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule → Katalog SWT		
11. Empfohlene Voraussetzungen:	Einführung in die Technische Informatik (14360)		
12. Lernziele:	Kenntnisse über die Grundlagen der Organisation von Rechnern und deren maschinennahe Programmierung Grundzüge über die Beschreibung und den Entwurf von Hardwaresystemen		
13. Inhalt:	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: <ul style="list-style-type: none"> • Grundstrukturen: Stack-, Akkumulator- und Register-basierende Rechner • Informationsdarstellung in Hardware: Daten- und Befehlsformate, Fehlererkennung und -korrektur, Befehlssätze und Unterstützung von Hochsprachen. • MIPS als RISC-Bespiel und seine Assemblerprogrammierung • Grundelemente und Entwurf auf Register-Transfer-Ebene • Grundzüge einer Hardware-Beschreibungssprache • Operationswerke: Multiplikation, Division, Gleitkommaeinheiten • Steuerwerksentwurf und Mikroprogrammierung • Befehlszyklus und Unterbrechungen • Pipelining und statisches Scheduling • Speicherorganisation: Cachestrukturen und virtueller Speicher, Seitenverwaltung, Segmentierung, TLB, MMU und DMA • Leistungsbewertung: Maßzahlen und CPI, Benchmarking und einfache Anwendung von Warteschlangen 		

14. Literatur:	Literatur, siehe Webseite zur Veranstaltung
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none">• 167901 Vorlesung Rechnerorganisation 1• 167902 Praktische und theoretische Übungen Rechnerorganisation 1• 365302 Praktische und theoretische Übungen Rechnerorganisation 1
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none">• 36531 Rechnerorganisation 1 (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0• V Vorleistung (USL-V), schriftliche Prüfung
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:	Prof.Dr. Kurt Rothermel		
9. Dozenten:	<ul style="list-style-type: none"> • Kurt Rothermel • Frank Leymann 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009</p> <ul style="list-style-type: none"> → Ergänzungsmodule → Katalog ISG <p>B.Sc. Softwaretechnik, PO 2009</p> <ul style="list-style-type: none"> → Ergänzungsmodule → Katalog ISW <p>B.Sc. Softwaretechnik, PO 2009</p> <ul style="list-style-type: none"> → Ergänzungsmodule → Katalog SWT <p>B.Sc. Softwaretechnik, PO 2012</p> <ul style="list-style-type: none"> → Ergänzungsmodule → Katalog ISG <p>B.Sc. Softwaretechnik, PO 2012</p> <ul style="list-style-type: none"> → Ergänzungsmodule → Katalog ISW <p>B.Sc. Softwaretechnik, PO 2012</p> <ul style="list-style-type: none"> → Ergänzungsmodule → Katalog SWT 		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> * Modul 051520005 Programmierung und Software-Entwicklung * Modul 051510005 Datenstrukturen und Algorithmen 		
12. Lernziele:	<ul style="list-style-type: none"> * Verstehen grundlegender Architekturen und Organisationsformen von Software-Systemen * Verstehen systemnaher Konzepte und Mechanismen * Kann existierende Systemplattformen und Betriebssysteme hinsichtlich ihrer Eigenschaften analysieren und anwenden. * Kann systemnahe Software entwerfen und implementieren. * Kann nebenläufige Programme entwickeln * Kann mit Experten anderer Fachgebiete die Anwendung von Systemfunktionen abstimmen. 		
13. Inhalt:	<p>Grundlegende Systemstrukturen - und organisationen</p> <ul style="list-style-type: none"> • Multitaskingsystem • Multiprozessorsystem • Verteiltes System <p>Modellierung und Analyse nebenläufiger Programme</p> <ul style="list-style-type: none"> • Abstraktionen: Atomare Befehle, Prozesse, nebenläufiges Programm • Korrektheit- und Leitungskriterien <p>Betriebssystemkonzepte</p> <ul style="list-style-type: none"> • Organisation von Betriebssystemen • Prozesse und Threads • Eingabe/Ausgabe • Scheduling 		

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

Modul: 11330 Visualisierung

2. Modulkürzel:	051900011	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.Dr. Daniel Weiskopf		
9. Dozenten:	<ul style="list-style-type: none"> • Thomas Ertl • Daniel Weiskopf • Filip Sadlo 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 6. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module</p> <p>M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • 051900002 Computergraphik • 10240 Numerische und Stochastische Grundlagen <i>oder</i>: • 41590 Einführung in die Numerik und Stochastik für Softwaretechniker 		
12. Lernziele:	Die Studierenden haben Wissen über Grundlagen, Algorithmen und Datenstrukturen für die Visualisierung sowie praktische Fähigkeiten durch die Arbeit mit Visualisierungssoftware erworben.		
13. Inhalt:	<p>Visualisierung behandelt alle Aspekte, die mit der visuellen Repräsentation von Daten aus wissenschaftlichen Experimenten, Simulationen, medizinischen Scannern, Datenbanken oder ähnlichen Datenquellen gewonnen werden, um zu einem tieferen Verständnis zu gelangen oder eine einfachere Darstellung komplexer Phänomene oder Sachverhalte zu erhalten. Um dieses Ziel zu erreichen, werden zum einen wohlbekannte Techniken aus dem Gebiet der interaktiven Computergraphik, zum anderen auch neu entwickelte Techniken angewendet.</p> <p>Entsprechend werden in dieser Vorlesung folgenden Themen behandelt:</p> <ul style="list-style-type: none"> • Einführung, Historie, Visualisierungspipeline • Datenakquise und -repräsentation (Abtasten, Rekonstruktion, Gitter, Datenstrukturen) • Wahrnehmungsaspekte • Grundlegende Konzepte visueller Abbildungen • Visualisierung von Skalarfeldern (Isoflächenextraktion, Volumenrendering) • Visualisierung von Vektorfelder (Teilchenverfolgung, texturbasierte Methoden, Topologie) • Tensorfelder, Multiattributdaten 		
14. Literatur:	<ul style="list-style-type: none"> • C. D. Hansen, C. R. Johnson, The Visualization Handbook, 2005 • C. Ware, Information Visualization: Perception for Design, 2004 • H. Schumann, W. Müller, Visualisierung: Grundlagen und allgemeine Methoden, 2000 		

	<ul style="list-style-type: none">• K. Engel, M. Hadwiger, J. M. Kniss, C. Rezk-Salama, D. Weiskopf, Real-time Volume Graphics, 2006
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none">• 113301 Vorlesung Visualisierung• 113302 Übungen Visualisierung
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none">• 11331 Visualisierung (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0• V Vorleistung (USL-V), schriftlich, eventuell mündlich, Erfolgreiche Teilnahme an Übungen / excercises passed.
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	

310 Katalog SWT

Zugeordnete Module:

- 10150 Grundlagen des Compilerbaus und der Programmiersprachen
- 10210 Mensch-Computer-Interaktion
- 10220 Modellierung
- 36530 Rechnerorganisation 1
- 40090 Systemkonzepte und -programmierung
- 41590 Einführung in die Numerik und Stochastik für Softwaretechniker

Modul: 41590 Einführung in die Numerik und Stochastik für Softwaretechniker

2. Modulkürzel:	051240006	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Dr. Dirk Pflüger		
9. Dozenten:	<ul style="list-style-type: none"> • Dirk Pflüger • Stefan Zimmer • Thomas Ertl • Daniel Weiskopf • Marc Alexander Schweitzer 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog SWT</p> <p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Katalog SWT</p>		
11. Empfohlene Voraussetzungen:	Mathematik für Informatiker und Softwaretechniker (Modulkürzel 080300100; 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"> • numerische Algorithmik • Gleitpunktzahlen und Gleitpunktarithmetik • Interpolation & Approximation • Integration • lineare Gleichungssysteme • Iterative Lösung linearer und nichtlinearer Gleichungen • gewöhnliche Differentialgleichungen • Stochastik 		

- Zufall und Unsicherheit
- diskrete und kontinuierliche Wahrscheinlichkeitsräume
- Asymptotik

14. Literatur:	<ul style="list-style-type: none">• Huckle, Schneider; Numerik für Informatiker• Schickinger T., Steger A.; Diskrete Strukturen, Band 2, 2002• Dahmen, Reusken; Numerik für Ingenieure
15. Lehrveranstaltungen und -formen:	
16. Abschätzung Arbeitsaufwand:	<ul style="list-style-type: none">• Präsenzzeit: 42 Stunden• Nachbearbeitungszeit: 138 Stunden
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: 10150 Grundlagen des Compilerbaus und der Programmiersprachen

2. Modulkürzel:	051510015	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Prof.Dr. Erhard Plödereder		
9. Dozenten:	Erhard Plödereder		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog SWT</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog SWT</p>		
11. Empfohlene Voraussetzungen:	<p>Grundkenntnisse aus den Einführungsvorlesungen des Informatikgrundstudiums, sowie einige Erfahrungen mit Programmierung. Vorkenntnisse über formale Sprachen sind vorteilhaft, aber nicht zwingend.</p>		
12. Lernziele:	<p>Die Studierenden haben die Grundkenntnisse erlangt, die zur effizienten Verwendung von Lexer- und Parser-Generatoren zur Analyse von Eingabetexten nötig sind. Sie verstehen die grundlegende Funktionsweise mehrerer Parse-Verfahren und kennen deren grammatikalischen Einschränkungen. Sie haben gelernt, die Fehlermeldungen aus diesen Generatoren und den Compilern oder Interpretern richtig einzuordnen. Ferner haben sie durch Betrachtung der Implementierungsmodelle typischer Programmiersprachenkonstrukte Verständnis für das Ausführungsverhalten und für typische, gefährliche Fehlerquellen in Anwendungsprogrammen erlangt.</p>		
13. Inhalt:	<p>Compilerarchitekturen im Überblick; lexikalische und syntaktische Analyse von Texten mit formaler Grammatik, insb. von Programmiersprachen. Lexikalische Analyse: endliche Automaten und ihre Implementierung; Syntaxanalyse: diverse Parser-Strategien, ihre Implementierung und Eigenschaften. Methoden der automatischen Generierung von Analysatoren aus Spezifikationen der Grammatiken. Fehlererkennung und -behandlung. Analyse der statischen Semantik: Grundbegriffe und elementare Methoden. Eigenschaften von Programmiersprachen; Realisierung der Laufzeitsemantik prozeduraler Programmiersprachen aus Benutzersicht, insbesondere Implementierungsmodelle der Speicherverwaltung und</p>		

der Unterprogrammaufrufe. Vermeidung typischer Fehlerquellen und überraschender Probleme in Anwendungsprogrammen.

(Nach SS14 wird sich der programmiersprachliche Teil ändern.)

14. Literatur:	<ul style="list-style-type: none">• Aho, Sethi, Ullman, Compilers - Principles, Techniques, and Tools, 1988• Wilhelm, Maurer, Uebersetzerbau, 1997
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none">• 101501 Vorlesung Grundlagen des Compilerbaus und der Programmiersprachen• 101502 Übung Grundlagen des Compilerbaus und der Programmiersprachen
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10151 Grundlagen des Compilerbaus und der Programmiersprachen (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	

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.Dr. Albrecht Schmidt		
9. Dozenten:	<ul style="list-style-type: none"> • Albrecht Schmidt • Thomas Ertl • Daniel Weiskopf 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog SWT</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog SWT</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • 051520005 Programmierung und Software-Entwicklung • 051200005 Systemkonzepte und -programmierung 		
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"> • Einführung in die Grundlagen der Mensch-Computer Interaktion, historische Entwicklung • Entwurfsprinzipien und Modelle für moderne Benutzungsschnittstellen und interaktive Systeme • Informationsverarbeitung des Menschen, Wahrnehmung, Motorik, Eigenschaften und Fähigkeiten des Benutzers 		

- Interaktionskonzepte und -stile, Metaphern, Normen, Regeln und Style Guides
- 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 Stunden
Nachbearbeitungszeit: 138 Stunden

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:	Prof.Dr. Frank Leymann		
9. Dozenten:	<ul style="list-style-type: none"> • Bernhard Mitschang • Frank Leymann 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2009, 4. Semester → Ergänzungsmodule → Katalog SWT</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012, 4. Semester → Ergänzungsmodule → Katalog SWT</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • 051520005 Programmierung und Software-Entwicklung • 051510005 Datenstrukturen und Algorithmen • 051200005 Systemkonzepte und -programmierung 		
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"> • Entity-Relationship Modell & komplexe Objekte • Relationenmodell & Relationenalgebra , Überblick SQL • Transformationen von ER nach Relationen, Normalisierung • XML, DTD, XML-Schema, Info-Set, Namensräume • Metamodelle & Repository • RDF, RDF-S & Ontologien • UML • Petri Netze, Workflownetze • BPMN 		
14. Literatur:	<ul style="list-style-type: none"> • A. Silberschatz, H. F. Korth, S. Sudarshan, Database System Concepts, 2002 • R. Eckstein, S. Eckstein, "XML und Datenmodellierung", dpunkt.verlag 2004 • M. Hitz, G. Kappel, E. Kapsammer, W. Retschitzegger, UML @ Work - Objektorientierte Modellierung mit UML2, 2005 • P. Hitzler, M. Krötzsch, S. Rudolph, Y. Sure, Semantic Web, 2008 • T.J. Teorey, Database Modeling & Design, 2nd Edition, 1994 		

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

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:	-
8. Modulverantwortlicher:	Prof.Dr. Hans-Joachim Wunderlich		
9. Dozenten:	Hans-Joachim Wunderlich		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog ISW B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog SWT B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule → Katalog ISW B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule → Katalog SWT		
11. Empfohlene Voraussetzungen:	Einführung in die Technische Informatik (14360)		
12. Lernziele:	Kenntnisse über die Grundlagen der Organisation von Rechnern und deren maschinennahe Programmierung Grundzüge über die Beschreibung und den Entwurf von Hardwaresystemen		
13. Inhalt:	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: <ul style="list-style-type: none"> • Grundstrukturen: Stack-, Akkumulator- und Register-basierende Rechner • Informationsdarstellung in Hardware: Daten- und Befehlsformate, Fehlererkennung und -korrektur, Befehlssätze und Unterstützung von Hochsprachen. • MIPS als RISC-Bespiel und seine Assemblerprogrammierung • Grundelemente und Entwurf auf Register-Transfer-Ebene • Grundzüge einer Hardware-Beschreibungssprache • Operationswerke: Multiplikation, Division, Gleitkommaeinheiten • Steuerwerksentwurf und Mikroprogrammierung • Befehlszyklus und Unterbrechungen • Pipelining und statisches Scheduling • Speicherorganisation: Cachestrukturen und virtueller Speicher, Seitenverwaltung, Segmentierung, TLB, MMU und DMA • Leistungsbewertung: Maßzahlen und CPI, Benchmarking und einfache Anwendung von Warteschlangen 		

14. Literatur:	Literatur, siehe Webseite zur Veranstaltung
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none">• 167901 Vorlesung Rechnerorganisation 1• 167902 Praktische und theoretische Übungen Rechnerorganisation 1• 365302 Praktische und theoretische Übungen Rechnerorganisation 1
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none">• 36531 Rechnerorganisation 1 (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0• V Vorleistung (USL-V), schriftliche Prüfung
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:	Prof.Dr. Kurt Rothermel		
9. Dozenten:	<ul style="list-style-type: none"> • Kurt Rothermel • Frank Leymann 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog SWT</p> <p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Katalog SWT</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> * Modul 051520005 Programmierung und Software-Entwicklung * Modul 051510005 Datenstrukturen und Algorithmen 		
12. Lernziele:	<ul style="list-style-type: none"> * Verstehen grundlegender Architekturen und Organisationsformen von Software-Systemen * Verstehen systemnaher Konzepte und Mechanismen * Kann existierende Systemplattformen und Betriebssysteme hinsichtlich ihrer Eigenschaften analysieren und anwenden. * Kann systemnahe Software entwerfen und implementieren. * Kann nebenläufige Programme entwickeln * Kann mit Experten anderer Fachgebiete die Anwendung von Systemfunktionen abstimmen. 		
13. Inhalt:	<p>Grundlegende Systemstrukturen - und organisationen</p> <ul style="list-style-type: none"> • Multitaskingsystem • Multiprozessorsystem • Verteiltes System <p>Modellierung und Analyse nebenläufiger Programme</p> <ul style="list-style-type: none"> • Abstraktionen: Atomare Befehle, Prozesse, nebenläufiges Programm • Korrektheit- und Leitungskriterien <p>Betriebssystemkonzepte</p> <ul style="list-style-type: none"> • Organisation von Betriebssystemen • Prozesse und Threads • Eingabe/Ausgabe • Scheduling 		

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

340 Wahlmodule aus Master SWT

Zugeordnete Module:	10040	Bildsynthese
	29430	Computer Vision
	29440	Geometric Modeling and Computer Animation
	29480	Loose Coupling and Message Based Applications
	29490	Services und Service Komposition
	29500	Visual Computing
	29520	Ausgewählte Themen des Service Computing
	29530	Business Process Management
	42520	Services and Service Composition
	46760	Theoretical and Methodological Foundations of Visual Computing
	55630	Information Visualization and Visual Analytics
	55640	Correspondence Problems in Computer Vision

Modul: 29520 Ausgewählte Themen des Service Computing

2. Modulkürzel:	052010005	5. Moduldauer:	2 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	5.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Prof.Dr. Frank Leymann		
9. Dozenten:	<ul style="list-style-type: none"> • Frank Leymann • Dimka Karastoyanova 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Wahlmodule aus Master SWT</p> <p>B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module</p> <p>M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT</p>		
11. Empfohlene Voraussetzungen:	Modul Web Services 1		
12. Lernziele:	Die Konzepte des Grid Computing und die Beziehungen zu Web Service Technologie werden erläutert. Die Grundlagen des Semantic Web werden vermittelt, und die wesentlichen Ansätze für Semantic Web Services werden erklärt.		
13. Inhalt:	<p>Zunächst wird die Web Services Technologien zusammengefasst und die Ansätze zum Orchestrieren von Web Services dargestellt, insbesondere die Workflow Technologie und die BPEL Sprache. Grid Services und Infrastrukturen werden diskutiert. Das Web Service Resource Framework wird ausführlich vorgestellt.</p> <p>Danach werden die Grundlagen des Semantic Web und der Semantic Web Services vorgestellt und die Bedeutung der Nutzung von Semantik wird diskutiert. Eine Übersicht über die wichtigsten existierenden Ontologien wird vorgestellt, sowie über die Benutzung einiger der existierenden Werkzeuge demonstriert. Die Grundlagen von Semantic Web Services und die Vorteile und Nachteile der drei existierenden Semantic Web Service Technologien (OWL-S, WSMO, SAWSDL) werden dargestellt. Der Einsatz von Semantik bei der Erstellung und Ausführung von Prozessen wird eingeführt, sowie einige existierende Ansätze dargestellt.</p>		
14. Literatur:	Thomas B. Passin: "Explorer's Guide to the Semantic Web", 2004 (Ressourcen im Web)		
15. Lehrveranstaltungen und -formen:			
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 60 Stunden Nachbearbeitungszeit: 120 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 29521 Ausgewählte Themen des Service Computing (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0 • V Vorleistung (USL-V), 		
18. Grundlage für ... :			
19. Medienform:	Vorlesung mit begleitenden Übungen		

20. Angeboten von:

Architektur von Anwendungssystemen

Modul: 10040 Bildsynthese

2. Modulkürzel:	051900012	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Prof.Dr. Thomas Ertl		
9. Dozenten:	<ul style="list-style-type: none"> • Martin Fuchs • Thomas Ertl • Daniel Weiskopf 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 6. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule → Wahlmodule aus Master SWT</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module</p> <p>M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • Modul 051900002 Computergraphik 		
12. Lernziele:	<p>Die Studierenden besitzen Wissen über verschiedene Ansätze und Algorithmen der dreidimensionalen Computergraphik, physikalisch-basierte Verfahren wie Raytracing und Radiosity, die den Lichttransport und die Wechselwirkung mit Materie modellieren, und numerische Methoden wie Monte-Carlo-Integration und Finite-Elemente-Verfahren die es erlauben, die Rendering-Gleichung zu lösen. Darüber hinaus kennen sie interaktive Verfahren, die unter Ausnutzung programmierbarer Grafik-Hardware realistische Beleuchtungseffekte in Echtzeit approximieren können, sowie bildbasierte Ansätze, die ohne geometrische Daten realistische Darstellungen erzeugen. Bild-basierte Verfahren verzichten auf eine geometrische Repräsentation der Szene und erzeugen neue Ansichten aus anderen aufgenommenen Bildern.</p>		
13. Inhalt:	<p>In dieser Vorlesung werden die folgenden Themen behandelt:</p> <ul style="list-style-type: none"> • Grafik Hardware und APIs, OpenGL • Texturen, prozedurale Modelle • Schattenberechnungen • Szenengraphen, Culling, Level-of-Detail Verfahren • Physikalisch-basierte Beleuchtungsberechnung, Fotorealistische Bildsynthese • Lokale Beleuchtungsmodelle • Raytracing, Monte-Carlo Methoden • Radiosity • Bild-basiertes Rendering 		
14. Literatur:	<ul style="list-style-type: none"> • Andrew S. Glassner, Principles of Digital Image Synthesis, 1995 • D. Eberly, 3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics, 2000 		

- J. Foley, A. van Dam, S. Feiner, J. Hughes, Computer Graphics: Principle and Practice, 1990
- Literatur, siehe Webseite zur Veranstaltung
- P. Dutre, P. Bekaert, K. Bala, Advanced Global Illumination, 2003
- Tomas Akenine-Möller, Eric Haines, Real-Time Rendering, 2002
- Matt Pharr, Greg Humphreys, Physically Based Rendering: From Theory To Implementation, Morgan Kaufmann; Auflage: 2nd revised edition. (26. August 2010)
- Peter Shirley et al, Fundamentals of Computer Graphics, Third Edition, A.K. Peters, July 2009

15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 100401 Vorlesung Bildsynthese • 100402 Übung Bildsynthese
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 10041 Bildsynthese (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0 • V Vorleistung (USL-V), schriftlich, eventuell mündlich, Übungsschein.
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme

Modul: 29530 Business Process Management

2. Modulkürzel:	052010006	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Prof.Dr. Frank Leymann		
9. Dozenten:	<ul style="list-style-type: none"> • Frank Leymann • Dimka Karastoyanova 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Wahlmodule aus Master SWT</p> <p>B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module</p> <p>M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT</p>		
11. Empfohlene Voraussetzungen:			
12. Lernziele:	<p>The concept of workflows (aka business processes) are understood. The formal basics of modelling and executing workflow models are clear. Concrete languages can used in principle. The architecture of workflow systems will be discussed. The lifecycle of business process management is clear.</p>		
13. Inhalt:	<p>Since as early as the 90's, business process re-engineering has been the top priority of businesses all over the globe. Seamlessly adapting the business processes of an enterprise to evolving markets, business strategies and unforeseen events is regarded as a vital capability by the business community at large.</p> <p>Workflows are the leading approach for specifying and automating business processes in enterprises. The course will provide an extensive insight on the relationship entwining business processes and workflows in the current practice of Business Process Management (BPM). The concepts surrounding workflows will be dissected, including workflow management systems, their mathematical foundations, transactional workflows and fault handling. The technological embodiment of workflows will be treated in the scope of Service-Oriented Architecture (SOA), a mainstay of BPM in the current practice of enterprises. In particular, the course will cover in detail the Business Process Execution Language (BPEL), the de-facto, industry-wide standard for automating business processes specified as executable workflows that leverage the technologies offered in the scope of SOA. Equally, the more business oriented modelling language BPMN (Business Process Model and Notation) will be introduced.</p>		
14. Literatur:	<p>F. Leymann, D. Roller: "Production Workflow", Prentice Hall 2000 M. Dumas, W. van der Aalst, A. ter Hofstede: "Process-Aware Information Systems", John Wiley & Sons 2005 M. Weske: "Business Process Management - Concepts, Languages, Architectures". Springer 2007 Bruce Silver: "BPMN Method & Style", Cody- Cassidy Press 2009</p>		
15. Lehrveranstaltungen und -formen:			
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 Stunden	

Nachbearbeitungszeit: 138 Stunden

17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none">• 29531 Business Process Management (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0• V Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	29540 Ausgewählte Themen des Business Process Management
19. Medienform:	Lecture with accompanying exercises
20. Angeboten von:	Architektur von Anwendungssystemen

Modul: 29430 Computer Vision

2. Modulkürzel:	051900215	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof.Dr.-Ing. Andrés Bruhn		
9. Dozenten:	Andrés Bruhn		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Wahlmodule aus Master SWT B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • Modul 080300100 Mathematik für Informatiker und Softwaretechniker • Modul 050700005 Imaging Science 		
12. Lernziele:	<p>Der Student / die Studentin beherrscht die Grundlagen der Merkmalsextraktion und -repräsentation, des 3-D Maschinensehens, der Bildsegmentierung sowie der Mustererkennung. Er/sie kann Probleme aus dem Fachgebiet einordnen und diese selbständig mit den erlernten Algorithmen und Verfahren lösen.</p> <p>The student knows the basics of feature extraction and representation, 3-D computer vision, image segmentation and pattern recognition. He/ she can solve problems of the field using the methods discussed in the course.</p>		
13. Inhalt:	<ul style="list-style-type: none"> • Lineare Diffusion, Skalenräume • Bildpyramiden, Kanten und Eckendetektion • Hough-Transformation, Invarianten • Texturanalyse • Scale Invariant Feature Transform (SIFT) • Bildfolgenanalyse: lokale Verfahren • Bewegungsmodelle, Objektverfolgung, Feature Matching • Bildfolgenanalyse: globale Verfahren • Kamerageometrie, Epipolargeometrie • Stereo Matching und 3-D Rekonstruktion • Shape-from-Shading • Isotrope und anisotrope nichtlineare Diffusion • Segmentierung mit globalen Verfahren • Kontinuierliche Morphologie, Schockfilter • Mean Curvature Motion • Self-Snakes, Aktive Konturen • Bayes'sche Entscheidungstheorie der Mustererkennung • Klassifikation mit parametrischen Verfahren, Dichteschätzung • Klassifikation mit nicht-parametrischen Verfahren • Dimensionsreduktion <ul style="list-style-type: none"> • Linear Diffusion, Scale Space • Image Pyramids, Edges and Corners • Hough Transform, Invariants • Texture Analysis • Scale Invariant Feature Transform 		

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

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

Modul: 55640 Correspondence Problems in Computer Vision

2. Modulkürzel:	051900211	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	6.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof.Dr.-Ing. Andrés Bruhn		
9. Dozenten:	Andrés Bruhn		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Wahlmodule aus Master SWT B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • Modul 080300100 Mathematik für Informatiker und Softwaretechniker • Modul 050700005 Imaging Science • Modul 051900215 Computer Vision 		
12. Lernziele:	<p>Der Student kann Korrespondenzprobleme im Computer-Vision-Bereich selbständig einordnen, Lösungsstrategien mathematisch modellieren und diese dann geeignet algorithmisch umsetzen.</p> <p>The student has knowledge on the different correspondence problems in computer vision, is able to develop mathematical models for solution strategies and implement the corresponding algorithms in an appropriate way.</p>		
13. Inhalt:	<ul style="list-style-type: none"> • Basisverfahren: Block Matching, Detektion von Verdeckungen, Merkmalsfindung, Feature Matching • Optischer Fluss: Lokale und Globale differentiale Verfahren, Parametrisierungsmodelle, Konstanzannahmen, Daten- und Glattheitsterme, Numerik, Große Verschiebungen, Hochgenaue Verfahren • Stereorekonstruktion: Projektive Geometrie, Epipolargeometrie, Schätzung der Fundamentalmatrix • Szenenfluss: Gemeinsame Schätzung von Struktur, Bewegung und Geometrie • Medizinische Bildregistrierung: Mutual Information, Elastische und krümmungsbasierte Regularisierung, Landmarks • Particle Image Velocimetry: Div-Curl-Regularisierung, Inkompressibler Navier Stokes Prior • Basic Approaches: Block Matching, Occlusion Detection, Interest Points, Feature Matching • Optic Flow: Local and Global Differential Methods, Parametrisation Models, Constancy Assumptions, Data and Smoothness Terms, Numerics, Large Displacements, High Accuracy Methods • Stereop Matching: Projective Geometry, Epipolar Geometry, Estimation of the Fundamental Matrix • Scene Flow: Joint Estimation of Structure, Motion, and Geometry • Medical Image Registration: Mutual Informaion, Elastic and Curvature-Based Regularisation, Landmarks 		

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

Modul: 29440 Geometric Modeling and Computer Animation

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

	<ul style="list-style-type: none"> • Fluid simulation: wave equation, Navier Stokes, level sets, particle level sets • Basics of film production: camera, lighting, production process, storyboard
14. Literatur:	<ul style="list-style-type: none"> • D. Eberly, 3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics. Morgan Kaufmann, 2000 • G. Farin: Curves and Surfaces for CAGD: A Practical Guide. Morgan Kaufmann, 2002 • R. Parent: Computer Animation: Algorithms and Techniques. Morgan Kaufmann, 2002 • W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling: Numerical Recipes - The Art of Scientific Computing. Cambridge University Press, 1986
15. Lehrveranstaltungen und -formen:	294401 Vorlesung mit Übungen Geometrische Modellierung und Animation
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden, Selbststudium: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 29441 Geometric Modeling and Computer Animation (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0, • V Vorleistung (USL-V), schriftlich, eventuell mündlich, Erfolgreiche Teilnahme an Übungen / exercises passed
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	

Modul: 55630 Information Visualization and Visual Analytics

2. Modulkürzel:	051900099	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof.Dr. Daniel Weiskopf		
9. Dozenten:	<ul style="list-style-type: none"> • Michael Burch • Thomas Ertl • Daniel Weiskopf 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Wahlmodule aus Master SWT</p> <p>B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module</p> <p>M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT</p>		
11. Empfohlene Voraussetzungen:	Basic Human Computer Interaction		
12. Lernziele:	Student gains expertise about fundamental concepts and techniques of information visualization and visual analytics. This includes algorithms and mathematical background, data structures and implementation aspects as well as practical experience with widely available visualization tools.		
13. Inhalt:	<p>Topics covered in this course:</p> <ul style="list-style-type: none"> - Perception and Cognition - Graphs and Networks - Hierarchies and Trees - Multi-dimensional and high-dimensional data visualization - Time series visualization - Visual Analytics - Software Visualization - Geospatial visualization 		
14. Literatur:	<ul style="list-style-type: none"> • Colin Ware. Visual Thinking for Design • Colin Ware. Information Visualization. Perception for Design • Edward Tufte. The Visual Display of Quantitative Information • Robert Spence. Design for Interaction • Jim Thomas. Illuminating the Path 		
15. Lehrveranstaltungen und -formen:	556301 Vorlesung und Übung Informationsvisualisierung		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden</p> <p>Gesamt: 180 Stunden</p>		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 55631 Information Visualization and Visual Analytics (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0 		

-
- V Vorleistung (USL-V), schriftlich, eventuell mündlich,
Erfolgreiche Übungsteilnahmen / excercises passed
-

18. Grundlage für ... :

19. Medienform: Tafel, Beamer, Matlab-Übungen auf PC

20. Angeboten von:

Modul: 29480 Loose Coupling and Message Based Applications

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

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

18. Grundlage für ... :

19. Medienform: Lecture and accompanying exercises

20. Angeboten von: Architektur von Anwendungssystemen

Modul: 42520 Services and Service Composition

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

Modul: 29490 Services und Service Komposition

2. Modulkürzel:	052010008	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Dr.-Ing. Dimka Karastoyanova		
9. Dozenten:	<ul style="list-style-type: none"> • Dimka Karastoyanova • Frank Leymann 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Wahlmodule aus Master SWT</p> <p>B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module</p> <p>M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT</p>		
11. Empfohlene Voraussetzungen:	021611 Grundlagen der Architektur von Anwendungssystemen, Vorlesung mit Übung, 4,0 SWS		
12. Lernziele:	Am Ende der Vorlesung ist das Konzept des Dienstes (Service) und der Lose Kopplung verstanden. SOA und der Zusammenhang mit Web Services sind klar. Die Methoden zur Entwicklung neuer Anwendungen aus Diensten sind verstanden und entsprechende Workflow-Sprachen können verwendet werden.		
13. Inhalt:	SOA und Lose Kopplung Dienstbeschreibungen (WSDL, Policy) Virtualisierung (Service Bus, etc.) Architektur von WFMS (Navigator, Executor, Worklist Manager, etc.) Flow Sprachen und Notationen (FDL, BPEL, BPMN, Petri Netze) Koordinationsprotokolle (WS-Coordination, WS-Transaction) Transaktionsunterstützung in Workflows		
14. Literatur:	<ul style="list-style-type: none"> • F. Leymann, D. Roller, Production Workflow, 2000 • Literatur, siehe Webseite zur Veranstaltung • S. Weerawarana, F. Curbera, F. Leymann, T. Storey, D. Ferguson, Web Services Platform Architecture, 2005 • W. van der Aalst, K. van Hee, Workflow Management, 2002 		
15. Lehrveranstaltungen und -formen:			
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 29491 Services und Service Komposition (PL), schriftlich oder mündlich, Gewichtung: 1.0 • V Vorleistung (USL-V), schriftlich, eventuell mündlich 		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 46760 Theoretical and Methodological Foundations of Visual Computing

2. Modulkürzel:	051900022	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof.Dr. Daniel Weiskopf		
9. Dozenten:	<ul style="list-style-type: none"> • Thomas Ertl • Daniel Weiskopf • Thomas Müller • Andrés Bruhn 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012</p> <ul style="list-style-type: none"> → Ergänzungsmodule → Wahlmodule aus Master SWT <p>B.Sc. Softwaretechnik, PO 2012</p> <ul style="list-style-type: none"> → Vorgezogene Master-Module <p>M.Sc. Softwaretechnik, PO 2012</p> <ul style="list-style-type: none"> → Spezialisierungsmodule - MSWT 		
11. Empfohlene Voraussetzungen:	<p>Module der Mathematik, Numerik und Stochastik aus dem BSc Informatik oder BSc Softwaretechnik:</p> <ul style="list-style-type: none"> • 10190 Mathematik für Informatiker und Softwaretechniker • 10240 Numerische und Stochastische Grundlagen <i>oder</i> • 41590 Einführung in die Numerik und Stochastik für Softwaretechniker 		
12. Lernziele:	<p>Die Studierenden kennen die mathematisch-theoretischen Grundlagen des Visual Computing und können diese in Form von Methoden für die Computergraphik, Visualisierung, Bildverarbeitung und Computer Vision praktisch umsetzen.</p>		
13. Inhalt:	<p>Die Vorlesung behandelt die Grundlagen der affinen und projektiven Geometrie und deren Umsetzung in der Computergraphik, insbesondere innerhalb der Grafikpipeline. Es wird die Differential- und Integralrechnung und deren Anwendung in zwei und drei Dimensionen behandelt. Grundlagen der Theorie der gewöhnlichen und partiellen Differentialgleichungen werden vermittelt. Interpolations- und Approximationsverfahren werden im Kontext von Visual Computing vertieft. Methoden der Fourier-Analyse sowie der diskreten Wavelet-Analyse und deren Anwendung in der Bildverarbeitung werden behandelt. Übungen vertiefen den theoretischen Vorlesungsstoff und dienen auch als praktische Einführung in die Umsetzung der Methoden für numerische Berechnungen und Algorithmen der Computergraphik, Visualisierung, Bildverarbeitung und Computer Vision.</p> <p>Im Einzelnen werden die folgenden Themen behandelt:</p> <ul style="list-style-type: none"> • Affine und projektive Geometrie: affiner Raum, affine Abbildung, orthographische und perspektivische Projektion, projektiver Raum, projektive Abbildung, homogene Koordinaten, Umsetzung in der Grafikpipeline • Differential- und Integralrechnung: partielle Ableitung, Gradient, Extrema in mehreren Variablen, numerische Ableitung, 		

- Kantendetektion, Taylor-Entwicklung in mehreren Variablen, vektorwertige Funktionen, Integralrechnung in mehreren Variablen
- Gewöhnliche und partielle Differentialgleichungen: Existenz und Eindeutigkeit, autonome Systeme, Vektorfelder, Integralkurven, numerische Verfahren
- Interpolation und Approximation: Lagrange-Interpolation, Interpolation höherer Ordnung, baryzentrische Koordinaten, radiale Basisfunktionen, Shepard, Moving Least Squares (MLS), Kriging
- Fourier-Analysis: kontinuierliche und diskrete Fourier-Transformation, Frequenz- und Phasenspektrum, Gibbs, Faltung, Dirac-delta, Abtasttheorem, diskrete Filter, Anwendungen in der Bildverarbeitung
- Wavelet-Transformation: Haar-Transformation und -Wavelet, Multiresolution-Analyse, Daubechies-Wavelets, Denoising, Bildverarbeitung
- Einführung in ein Softwaresystem zur praktischen Umsetzung (z.B. Matlab)

14. Literatur:

- B. Jähne. Digitale Bildverarbeitung. Springer, 2005
- H. Fischer, H. Kaul. Mathematik für Physiker - Band 1: Grundkurs. 5. Auflage, Teubner, 2005
- H. Fischer, H. Kaul. Mathematik für Physiker - Band 2: Gewöhnliche und partielle Differentialgleichungen, mathematische Grundlagen der Quantenmechanik. 2. Auflage, Teubner, 2004
- H. R. Schwarz, N. Köckler. Numerische Mathematik. 6. Auflage, Teubner, 2006
- J. S. Walker. A primer on WAVELETS and Their Scientific Applications. Chapman & Hall/CRC, 2008
- M. Oberguggenberger, A. Ostermann. Analysis für Informatiker. Springer, 2009
- J. Encarnaçao, W. Straßer, R. Klein. Graphische Datenverarbeitung 1. Oldenburg Verlag, 1996

15. Lehrveranstaltungen und -formen:

16. Abschätzung Arbeitsaufwand:

Präsenzzeit: 42 Stunden, Selbststudium: 138 Stunden

17. Prüfungsnummer/n und -name:

- 46761 Theoretical and Methodological Foundations of Visual Computing (PL), schriftlich, eventuell mündlich, 120 Min., Gewichtung: 1.0, schriftlich 120 Min. oder mündlich 30 Min.
- V Vorleistung (USL-V), schriftlich, eventuell mündlich, Erfolgreiche Teilnahme an Übungen

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

Modul: 29500 Visual Computing

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

15. Lehrveranstaltungen und -formen:	295001 Vorlesung mit Übungen Visual Computing
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden
	Selbststudium: 138 Stunden
	Gesamt: 180 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none">• 29501 Visual Computing (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0, Schriftliche Prüfung von 120 Min. oder mündlichen 30 Min• V Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	

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:	unregelmäßig
4. SWS:	10.0	7. Sprache:	-
8. Modulverantwortlicher:	Univ.-Prof.Dr. 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 Stunden Nachbearbeitungszeit: 210 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 16611 Studienprojekt-Pr (LBP), schriftlich oder 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. • V Vorleistung (USL-V), schriftlich, eventuell mündlich, 30 Min. • V Vorleistung (USL-V), schriftlich, eventuell mündlich, 30 Min. 		

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:
