



Universität Stuttgart

Modulhandbuch
Studiengang Bachelor of Arts (Kombination) Informatik
Prüfungsordnung: 2009

Universität Stuttgart
Keplerstr. 7
70174 Stuttgart

Inhaltsverzeichnis

Präambel	3
100 Module im Nebenfach	4
23530 Automaten und Formale Sprachen	5
12060 Datenstrukturen und Algorithmen	6
14360 Einführung in die Technische Informatik	8
320 Katalog ISG	10
10030 Architektur von Anwendungssystemen	11
10080 Datenbanken und Informationssysteme	13
10090 Grundlagen der Eingebetteten Systeme	15
10100 Grundlagen der Graphischen Ingenieursysteme	16
10110 Grundlagen der Künstlichen Intelligenz	18
10120 Grundlagen der Modellbildung und Simulation	20
10140 Grundlagen der Rechnerarchitektur	22
10150 Grundlagen des Compilerbaus und der Programmiersprachen	24
10170 Imaging Science	26
10250 Parallele Systeme	28
11110 Verteilte Systeme	29
330 Katalog ISW	31
10040 Bildsynthese	32
10050 Bildverstehen	34
10060 Computergraphik	36
11900 Design and Test of Systems on a Chip	38
14380 Hardware Verification and Quality Assessment	40
10160 Hardwarebeschreibungssprachen	42
10180 Information Retrieval und Text Mining	44
10300 Rechnernetze	45
11330 Visualisierung	47
10260 Programmierkurs	49
10280 Programmierung und Software-Entwicklung	51

Präambel

Die Studien- und Prüfungsordnung für den Bachelorstudiengang **Informatik** beschreibt den Aufbau des Studiums und die Organisation der Prüfungen. Sie stellt das Regelwerk und die Rechtsgrundlage für eine einheitliche Handhabung des Studienablaufs und der Bewertung der Studien- und Prüfungsleistungen dar. Sie wendet sich dabei sowohl an die Studierenden als auch an die Prüfenden sowie an die entsprechenden Organe der Universität Stuttgart.

Aus Gründen der sprachlichen Vereinfachung wird grundsätzlich nur die grammatikalisch männliche Form für Personen verwendet. Sinngemäß ist stets auch die entsprechende weibliche Form gemeint.

100 Module im Nebenfach

Zugeordnete Module:

- 23530 Automaten und Formale Sprachen
- 12060 Datenstrukturen und Algorithmen
- 14360 Einführung in die Technische Informatik
- 320 Katalog ISG
- 330 Katalog ISW
- 10260 Programmierkurs
- 10280 Programmierung und Software-Entwicklung

Modul: 23530 Automaten und Formale Sprachen

2. Modulkürzel:	050420007	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:

9. Dozenten:

10. Zuordnung zum Curriculum: BA (Komb) Informatik, 4. Semester
 → Module im Nebenfach

11. Voraussetzungen:

12. Lernziele:

13. Inhalt:

14. Literatur:

15. Lehrveranstaltungen und -formen: • 235301 Vorlesung Automaten und Formale Sprachen
 • 235302 Gruppenübungen Automaten und Formale Sprachen

16. Abschaetzung Arbeitsaufwand:

17a. Studienleistung:

17b. Prüfungsleistungen:

18. Grundlage für ... :

19. Medienform:

20. Prüfungsnummer/n und -name: 23531 Automaten und Formale Sprachen

21. Angeboten von:

22. Zuordnung zu weiteren Curricula:

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:	Erhard Plödereder		
9. Dozenten:	Stefan Funke		
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 2. Semester → Basismodule BA (Komb) Informatik, 2. Semester → Module im Nebenfach		
11. 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 		

	• 120602 Übung Datenstrukturen und Algorithmen
16. Abschaetzung Arbeitsaufwand:	Präsenzzeit: 63 Stunden Nachbearbeitungszeit: 207 Stunden
17a. Studienleistung:	Prüfungsvorleistung: Übungsschein.
17b. Prüfungsleistungen:	Schriftliche Prüfung von 120 Minuten Dauer. Die genauen Details der Übungsleistungen und Ihrer Anrechnung werden zu Beginn der Lehrveranstaltung bekannt gegeben.
18. Grundlage für ... :	
19. Medienform:	
20. Prüfungsnummer/n und -name:	12061 Datenstrukturen und Algorithmen
21. Angeboten von:	
22. Zuordnung zu weiteren Curricula:	B.Sc. Mathematik, 2. Semester → Nebenfach → Nebenfach Informatik B.Sc. Softwaretechnik, 2. Semester → Basismodule B.Sc. Wirtschaftsinformatik, 2. Semester → Informatik (B 1) B.Sc. Mechatronik, 2. Semester → Kernmodule B.Sc. Maschinelle Sprachverarbeitung, 2. Semester → Basismodule B.Sc. Technikpädagogik, 2. Semester → Hauptfach Informatik → Basismodule Informatik B.Sc. Technikpädagogik, 2. Semester → Wahlpflichtfach → Informatik M.Sc. Technikpädagogik, 2. Semester → Wahlpflichtfach B → Affines Wahlpflichtfach Informatik → Affines Wahlpflichtfach Informatik Basismodule

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:	Otto Eggenberger		
9. Dozenten:	Otto Eggenberger		
10. Zuordnung zum Curriculum:	BA (Komb) Informatik, 3. Semester → Module im Nebenfach		
11. 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:	<p>Grundlegende Funktionsweise eines Computers</p> <ul style="list-style-type: none"> • Informationsdarstellung • Zahlendarstellung und Codes • Digitale Grundbausteine • Logische Funktionen, Speicherelemente • Befehlsausführung, Programmablauf <p>Elektrotechnische Grundlagen</p> <ul style="list-style-type: none"> • Physikalische Grundbegriffe • Elektrische Spannung, elektrischer Strom • Elektrische Netzwerke • Halbleiterbauelemente • Digitale Grundschaltungen <p>Digitale Schaltungen</p> <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. Abschaetzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 126 Stunden		
17a. Studienleistung:	keine		
17b. Prüfungsleistungen:	Schriftliche Prüfung von 90 Minuten Dauer		
18. Grundlage für ... :			
19. Medienform:			

20. Prüfungsnummer/n und -name: 14361 Einführung in die Technische Informatik

21. Angeboten von:

22. Zuordnung zu weiteren Curricula: B.Sc. Softwaretechnik, 3. Semester
→ Kernmodule

320 Katalog ISG

Zugeordnete Module:	10030	Architektur von Anwendungssystemen
	10080	Datenbanken und Informationssysteme
	10090	Grundlagen der Eingebetteten Systeme
	10100	Grundlagen der Graphischen Ingenieursysteme
	10110	Grundlagen der Künstlichen Intelligenz
	10120	Grundlagen der Modellbildung und Simulation
	10140	Grundlagen der Rechnerarchitektur
	10150	Grundlagen des Compilerbaus und der Programmiersprachen
	10170	Imaging Science
	10250	Parallele Systeme
	11110	Verteilte Systeme

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:	Frank Leymann		
9. Dozenten:	Frank Leymann		
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 BA (Komb) Informatik, 4. Semester → Module im Nebenfach → Katalog ISG		
11. Voraussetzungen:	Vorlesungen des Grundstudiums.		
12. Lernziele:	Am Ende des Moduls ist das Konzept einer Anwendungsarchitektur verstanden. Die Rolle von Middleware im Rahmen von Anwendungsarchitekturen ist klar. Grundsätzliche Strukturen und Muster in Anwendungsarchitekturen sind bekannt. Nicht-funktionale Eigenschaften von Anwendungssystemen und deren Bedeutung sind verstanden.		
13. Inhalt:	<ul style="list-style-type: none"> • "Anwendungsarchitektur": Terminologie, Rollen & Artefakte • Datenbanksysteme • Schichtenarchitekturen (Client-Server Cut, N-Tier) • API, RPC & Middleware • Transaktionen • Nachrichtenorientierung • Message-Oriented Middleware • TP Monitore (direct TP, queued TP) • QoS (high availability, scalability, security,...) • Komponenten & ihre Beschreibung (EJB & WSDL) • Application Server (JEE) • Architekturstile (POSA,...) • Lose Kopplung & SOA • WfMS: Programmieren im Grossen • Optional: Lizenzierungen, Portabilität, Deployment, Konfiguration, Installation 		
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 		

Modul: 10080 Datenbanken und Informationssysteme

2. Modulkürzel:	051200025	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Bernhard Mitschang		
9. Dozenten:	<ul style="list-style-type: none"> • Bernhard Mitschang • Holger Schwarz 		
10. Zuordnung zum Curriculum:	<p>B.Sc. Informatik, 3. Semester</p> <ul style="list-style-type: none"> → Ergänzungsmodule → Katalog ISG 1-3 <p>BA (Komb) Informatik, 5. Semester</p> <ul style="list-style-type: none"> → Module im Nebenfach → Katalog ISG 		
11. Voraussetzungen:	<ul style="list-style-type: none"> • Modul 052010001 Modellierung 		
12. Lernziele:	Die Studierenden haben die erforderlichen Kenntnisse für Datenbankprogrammierer in angemessenem Umfang erworben.		
13. Inhalt:	<p>Diese Vorlesung ist als Einstiegsvorlesung für das Fachgebiet Datenbanksysteme konzipiert. Es wird dabei versucht, das Fachgebiet so gründlich und umfassend darzustellen, wie es für den Datenbankprogrammierer erforderlich und angemessen erscheint.</p> <p>Stoffauswahl, -umfang und Detaillierungsgrad wurden deshalb aus der Sicht der Anwendung von Datenbanksystemen getroffen, wozu hauptsächlich das Verständnis von Datenmodellen, der Entwurf von logischen Datenbankstrukturen (DB-Schemata) und der Umgang mit Datenbanksprachen gehören. Weiterhin soll durch Stoffauswahl das Verständnis anderer Vorlesungen, die gewisse Querbezüge und Verbindungen zu Datenbanksystemen aufweisen, erleichtert werden. Dazu gehören zum Beispiel Vorlesungen über Informationssysteme, Systemanalyse, Wissensdarstellung, Expertensysteme, Multimedia-Datenbanksysteme oder Rechnergestützte Ingenieursysteme (CAD/CAM).</p> <p>Übersicht:</p> <ul style="list-style-type: none"> • Informationsmodelle (Entity-Relationship-Modell) • Datenmodelle (Relationenmodell, Netzwerk- und hierarchisches Datenmodell) • Relationale Anfragesprachen (SQL, Relationenalgebra und Relationenkalkül) • Logischer DB-Entwurf (Relationensynthese und Normalformen) • Netzwerk-Datenmodell und Hierarchisches Datenmodell <p>Stoffauswahl, -umfang und Detaillierungsgrad werden aus der Sicht der Anwendung von Datenbanksystemen getroffen.</p>		
14. Literatur:	<ul style="list-style-type: none"> • A. Kemper, A. Eickler, Datenbanksysteme - Eine Einführung, 2004 • C. Date, An Introduction to Database Systems, 2003 • H. Garcia-Molina, J. D. Ullman, J. Widom, Database Systems. The Complete Book, 2003 • R. Elmasri, S. Navathe, Fundamentals of Database Systems, 2003 		

15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none">• 100801 Vorlesung Grundlagen der Datenbanken und Informationssysteme• 100802 Übung Grundlagen der Datenbanken und Informationssysteme
16. Abschaetzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17a. Studienleistung:	Prüfungsvorleistung; Modalitäten werden in der ersten Vorlesung angegeben
17b. Prüfungsleistungen:	Schriftliche Prüfung von 60 Minuten Dauer.
18. Grundlage für ... :	
19. Medienform:	
20. Prüfungsnummer/n und -name:	10081 Datenbanken und Informationssysteme
21. Angeboten von:	
22. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 3. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Maschinelle Sprachverarbeitung, 5. Semester → Wahlbereich E/I

Modul: 10090 Grundlagen der Eingebetteten Systeme

2. Modulkürzel:	051711010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Martin Radetzki		
9. Dozenten:	Martin Radetzki		
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISG		
11. Voraussetzungen:	<ul style="list-style-type: none"> • Modul 051700005 Rechnerorganisation 		
12. Lernziele:	Methodische und werkzeunterstützte Konstruktion von Hardware und Software für auf spezifische Anwendungen hin optimierte eingebettete Systeme.		
13. Inhalt:	Modelle zur Systemspezifikation, Modellierungssprachen und ihre Simulation, System- und Architektursynthese, Allokation von Ressourcen und Bindung von Aufgaben/Operationen, Hardware-Software-Partitionierung, Verfahren zur Ablaufplanung für parallele Architekturen, Optimierungsverfahren, Anwendungsspezifische Prozessoren, On-Chip/Board-Verbindungsnetzwerke.		
14. Literatur:	<ul style="list-style-type: none"> • 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. Abschaetzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17a. Studienleistung:	Keine.		
17b. Prüfungsleistungen:	Bewertete Rechnerübungen - vorlesungsbegleitend (0.25) Schriftliche Prüfung von 90 Minuten Dauer (0.75)		
18. Grundlage für ... :			
19. Medienform:			
20. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 10091 Grundlagen der Eingebetteten Systeme • 10092 Grundlagen der Eingebetteten Systeme - Rechnerübungen 		
21. Angeboten von:			
22. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW		

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:	Dieter Roller		
9. Dozenten:	<ul style="list-style-type: none"> • • Dieter Roller 		
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 BA (Komb) Informatik, 4. Semester → Module im Nebenfach → Katalog ISG		
11. 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. Abschaetzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17a. Studienleistung:	keine		
17b. Prüfungsleistungen:	Schriftliche Prüfung von 60 Minuten Dauer.		
18. Grundlage für ... :			
19. Medienform:			
20. Prüfungsnummer/n und -name:	10101 Grundlagen der Graphischen Ingenieursysteme		
21. Angeboten von:			
22. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 6. Semester → Ergänzungsmodule		

- Katalog ISG
 - B.Sc. Softwaretechnik, 6. Semester
 - Ergänzungsmodule
 - Katalog ISW
-

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:	Gunther Heidemann		
9. Dozenten:	Gunther Heidemann		
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISG		
11. 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 • Lernen 		
14. Literatur:	<ul style="list-style-type: none"> • G. F. Luger, Künstliche Intelligenz, 2001 • S. Russell, P. Norvig, Künstliche Intelligenz, 2004 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 101101 Vorlesung Grundlagen der Künstlichen Intelligenz • 101102 Übung Grundlagen der Künstlichen Intelligenz 		
16. Abschaetzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17a. Studienleistung:	Prüfungsvorleistung: Übungsschein		
17b. Prüfungsleistungen:	Schriftliche Prüfung von 60 Minuten Dauer.		
18. Grundlage für ... :			
19. Medienform:			
20. Prüfungsnummer/n und -name:	10111 Grundlagen der Künstlichen Intelligenz		
21. Angeboten von:			
22. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISG		

- B.Sc. Softwaretechnik, 5. Semester
 - Ergänzungsmodule
 - Katalog ISW
 - B.Sc. Wirtschaftsinformatik, 5. Semester
 - Informatik (B 1)
 - B.Sc. Wirtschaftsinformatik, 5. Semester
 - Wahlpflichtbereich (Bereich C)
 - Wahlbereich II: Informatik Basis
 - B.Sc. Wirtschaftsinformatik, 5. Semester
 - Wahlpflichtbereich (Bereich C)
 - Wahlbereich III: Informatik Simulation
 - B.Sc. Maschinelle Sprachverarbeitung, 5. Semester
 - Wahlbereich E/I
-

Modul: 10120 Grundlagen der Modellbildung und Simulation

2. Modulkürzel:	051240010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Marc Alexander Schweitzer		
9. Dozenten:	<ul style="list-style-type: none"> • Stefan Zimmer • Michael Bader • Marc Alexander Schweitzer 		
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 BA (Komb) Informatik, 4. Semester → Module im Nebenfach → Katalog ISG		
11. Voraussetzungen:	<ul style="list-style-type: none"> • 080300100 Mathematik für Informatiker und Softwaretechniker • 051240005 Numerische und Stochastische Grundlagen der Informatik 		
12. Lernziele:	Beherrschung des grundsätzlichen Vorgehens in der Modellbildung. Kenntnis einer Auswahl diskreter und kontinuierlicher Modelle und entsprechender Simulationsmethoden. Fähigkeit, mit den erlernten Kenntnissen selbständig numerische Methoden problemorientiert um- und einzusetzen.		
13. Inhalt:	Diese Vorlesung bietet eine Einführung in die Grundlagen der Modellbildung und Simulation mit dem Ziel der Vorbereitung auf weiterführende Vorlesungen in diesem Bereich. Da Simulationsmethoden oft für viele verschiedene Problemklassen einsetzbar sind, ist die Vorlesung methodisch strukturiert. Den Hauptteil der Vorlesung bilden hierbei kontinuierliche Modelle sowie deren numerische Behandlung. Vorgestellt werden insbesondere die Diskretisierung gewöhnlicher und partieller Differentialgleichungen mit Finiten Differenzen sowie die Lösung großer dünnbesetzter Gleichungssysteme, Eigenwert- und Minimierungsprobleme. Diese Verfahren werden auf Problemstellungen der Natur- und Ingenieurwissenschaften angewendet, z.B. Populationwachstum, Wärmetransport und Verkehrssimulation.		
14. Literatur:	<ul style="list-style-type: none"> • Modellbildung und Simulation - Eine anwendungsorientierte Einführung Bungartz, H.-J., Zimmer, S., Buchholz, M., Pflüger, D. , Springer Verlag, eXamen.press, 2009, ISBN 978-3-540-79809-5 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 101201 Vorlesung Grundlagen der Modellbildung und Simulation • 101202 Übung Grundlagen der Modellbildung und Simulation 		
16. Abschaetzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17a. Studienleistung:	Keine		
17b. Prüfungsleistungen:	Schriftliche Prüfung von 90 Minuten Dauer; bei geringer Teilnehmerzahl mündliche Prüfung von 30 Minuten Dauer.		
18. Grundlage für ... :			

19. Medienform:

20. Prüfungsnummer/n und -name: 10121 Grundlagen der Modellbildung und Simulation

21. Angeboten von:

22. Zuordnung zu weiteren Curricula:

- B.Sc. Softwaretechnik, 4. Semester
 - Ergänzungsmodule
 - Katalog ISG
- B.Sc. Softwaretechnik, 4. Semester
 - Ergänzungsmodule
 - Katalog ISW
- B.Sc. Softwaretechnik, 4. Semester
 - Ergänzungsmodule
 - Katalog SWT

Modul: 10140 Grundlagen der Rechnerarchitektur

2. Modulkürzel:	051700010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Hans-Joachim Wunderlich		
9. Dozenten:	Hans-Joachim Wunderlich		
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISG		
11. Voraussetzungen:	<ul style="list-style-type: none"> • 051700005 Rechnerorganisation 		
12. Lernziele:	<ul style="list-style-type: none"> • Grundlegendes Verständnis der Herausforderungen bei Einsatz, Entwurf und Fertigung moderner Prozessoren • Kenntnis über den Zusammenhang von Kosten, Performanz und Verlustleistung beim Entwurf von Prozessoren und der Konzeption komplexer Systeme • Kenntnis und Bewertung aktueller Architekturkonzepte 		
13. Inhalt:	<p>Es werden die klassischen Themen der Rechnerarchitektur als Hardware/Software-Schnittstelle sowie weiterführende technologische Themen behandelt.</p> <p>Dazu gehören insbesondere:</p> <ul style="list-style-type: none"> • Technologische Grundlagen: Entwurstile und Fertigungstechnik, Ausbeute, Test und Zuverlässigkeit, Kosten und Qualität, Skalierung. • Performanz: Taktfrequenz und Befehle pro Takt, Geschwindigkeitsanalyse und -optimierung. • Verlustleistung: Verlustleistungsanalyse, Optimierung von Verlustleistung und Performanz, Verlustleistung und Skalierung. • Computerarithmetik: Effiziente Hardwareimplementierung der Grundrechenarten, Hardwareimplementierung spezieller Funktionen wie Exponentialfunktion, Logarithmus und trigonometrische Funktionen, Gleitkommaoperationen und Standards, arithmetische Pipelines und Filter, aktuelle Gleitkommaeinheiten wie SPE des Cell Prozessors oder SPARC. • Parallelität auf Befehlsebene: Superskalarrechner, statisches und dynamisches Scheduling, Out-of-Order Ausführung und VLIW-Rechner, Multithreading • Parallele Architekturen: Shared Memory und Message Passing, Multi-Core Prozessoren und Multi-Core Systeme auf einem Chip. • Speicherhierarchie: Speichertechnologie und Cacheentwurf. • Fehlertoleranztechniken: Einzelprozessoren und Mehrprozessorsysteme. 		
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, 2006 • S. Iman, M. Pedram, Logic Synthesis for Low Power VLSI Designs, 1997 		

15. Lehrveranstaltungen und -formen: • 101401 Vorlesung Grundlagen der Rechnerarchitektur
 • 101402 Übung Grundlagen der Rechnerarchitektur

16. Abschaetzung Arbeitsaufwand: Präsenzzeit: 42 Stunden
 Nachbearbeitungszeit: 138 Stunden

17a. Studienleistung: Keine

17b. Prüfungsleistungen: Schriftliche Prüfung von 90 Minuten Dauer.

18. Grundlage für ... :

19. Medienform:

20. Prüfungsnummer/n und -name: 10141 Grundlagen der Rechnerarchitektur

21. Angeboten von:

22. Zuordnung zu weiteren Curricula: B.Sc. Softwaretechnik, 5. Semester
 → Ergänzungsmodule
 → Katalog ISG

 B.Sc. Softwaretechnik, 5. Semester
 → Ergänzungsmodule
 → Katalog ISW

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:	Erhard Plödereder		
9. Dozenten:	Erhard Plödereder		
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISG		
11. Voraussetzungen:	Grundkenntnisse aus den Einführungsvorlesungen des Informatikgrundstudiums, sowie einige Erfahrungen mit Programmierung. Vorkenntnisse über formale Sprachen sind vorteilhaft, aber nicht zwingend.		
12. Lernziele:	Die Studierenden haben die Grundkenntnisse erlangt, die zur effizienten Verwendung von Lexer- und Parser-Generatoren zur Analyse von Eingabetexten nötig sind. 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.		
13. Inhalt:	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 der Unterprogrammaufrufe. Vermeidung typischer Fehlerquellen und überraschender Probleme in Anwendungsprogrammen.		
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. Abschaetzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17a. Studienleistung:	Keine		

17b. Prüfungsleistungen:	Studienbegleitende Prüfung
18. Grundlage für ... :	
19. Medienform:	
20. Prüfungsnummer/n und -name:	10151 Grundlagen des Compilerbaus und der Programmiersprachen
21. Angeboten von:	
22. Zuordnung zu weiteren Curricula:	<ul style="list-style-type: none">B.Sc. Softwaretechnik, 4. Semester<ul style="list-style-type: none">→ Ergänzungsmodule→ Katalog ISGB.Sc. Softwaretechnik, 4. Semester<ul style="list-style-type: none">→ Ergänzungsmodule→ Katalog ISWB.Sc. Softwaretechnik, 4. Semester<ul style="list-style-type: none">→ Ergänzungsmodule→ Katalog SWT

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:	Gunther Heidemann		
9. Dozenten:	<ul style="list-style-type: none"> • Thomas Ertl • Daniel Weiskopf • Gunther Heidemann 		
10. Zuordnung zum Curriculum:	<p>B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3</p> <p>B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISW 1-3</p> <p>BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISG</p>		
11. Voraussetzungen:	<ul style="list-style-type: none"> • 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, Bildformate (z.B. ppm, gif, jpeg) • 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. AVI, MPEG) • Bildverbesserung und Restauration • Elementare Segmentierungsverfahren:Histogramme, Farben, Konturen • Elementare Mustererkennung <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, image file formats (e.g. ppm, gif, jpeg) • Basics of image processing, e.g. point operations such as contrast enhancement or binarization • Linear and nonlinear filtering such as convolution and morphological operations. 		

	<ul style="list-style-type: none"> • 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. avi, mpeg) • Image enhancement and restauration • Basics of segmentation: Histograms, colors, contours • Basics of pattern recognition
14. Literatur:	<ul style="list-style-type: none"> • Bässmann, Henning; Kreyss, 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 • Jähne, Bernd, Digitale Bildverarbeitung / Digital Image Processing, 2005 • 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. Abschaetzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17a. Studienleistung:	Prüfungsvorleistung: Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben
17b. Prüfungsleistungen:	Schriftliche Prüfung von 120 Minuten Dauer
18. Grundlage für ... :	
19. Medienform:	
20. Prüfungsnummer/n und -name:	10171 Imaging Science
21. Angeboten von:	
22. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW

Modul: 10250 Parallele Systeme

2. Modulkürzel:	051200065	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISW 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISG		
11. Voraussetzungen:	Ein Fach aus dem Bereich Technische Informatik.		
12. Lernziele:	Grundlegende Kenntnisse im Bereich paralleler Systeme, z.B. Multi-Core CPUs und deren Programmierung.		
13. Inhalt:	<ul style="list-style-type: none"> • Die Entwicklung vom klassischen Mikroprozessor zur Multi-Core CPU • Programmierung paralleler Rechnersysteme • Systolische Arrays, massiv parallele Systeme • Parallele Systeme aus verschiedenen • Anwendungsdomänen: ausgewählte Fallbeispiele 		
14. Literatur:	<ul style="list-style-type: none"> • Literatur, siehe Webseite zur Veranstaltung 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 102501 Vorlesung Parallele Systeme • 102502 Übung Parallele Systeme 		
16. Abschaetzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17a. Studienleistung:	Keine		
17b. Prüfungsleistungen:	Schriftliche Prüfung von 90 Minuten Dauer.		
18. Grundlage für ... :			
19. Medienform:			
20. Prüfungsnummer/n und -name:	10251 Parallele Systeme		
21. Angeboten von:			
22. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 3. Semester → Ergänzungsmodule → Katalog ISW		

Modul: 11110 Verteilte Systeme

2. Modulkürzel:	051200015	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Kurt Rothermel		
9. Dozenten:	Kurt Rothermel		
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISG		
11. Voraussetzungen:	<ul style="list-style-type: none"> • 051520005 Programmierung und Software-Entwicklung • 051510005 Datenstrukturen und Algorithmen • 051200005 Systemkonzepte und -Programmierung 		
12. Lernziele:	<ul style="list-style-type: none"> • Verstehen der grundsätzlichen Eigenschaften, Konzepte und Verfahren verteilter Systeme. • Kann existierende verteilte Anwendungen und Systemplattformen hinsichtlich ihrer Eigenschaften analysieren und verstehen. • Kann verteilte Anwendungen/Systemplattformen auf der Grundlage der erlernten Methoden realisieren. • Kann sich mit Experten anderer Fachdisziplinen über die Anwendung verteilter Systeme verständigen. 		
13. Inhalt:	<ul style="list-style-type: none"> • Einführung in die verteilten Systeme • Systemmodelle • Kommunikation: Nachrichten, Remote Procedure Call (RPC), Remote Method Invocation RMI) • Namensgebung: Generierung und Resolution • Zeit und Uhren in verteilten Systemen: Anwendungen, logische Uhren, physikalische Uhren, Uhrensynchronisation • Prozesssynchronisation: Wechselseitiger Ausschluß • Globaler Zustand: Konzepte, Snapshot Algorithmus, verteiltes Debugging • Transaktionsmanagement: Serialisierbarkeit, Sperrverfahren, 2-Phasen-Commit-Protokolle • Datenreplikation: Primary Copy, Consensus-Protokolle und andere Algorithmen • Sicherheit: Verfahren zur Geheimhaltung, Integrität, Authentifikation und Autorisierung • Broadcast-Algorithmen: Verarbeitungsmodell, Broadcast-Semantiken und -Algorithmen 		
14. Literatur:	Literatur, siehe Webseite zur Veranstaltung		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 111101 Vorlesung Verteilte Systeme • 111102 Übungen Verteilte Systeme 		
16. Abschaetzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiumszeit / Nachbearbeitungszeit:	138 h	
	Gesamt:	180 h	

17a. Studienleistung:	Keine
17b. Prüfungsleistungen:	Schriftlich Prüfung von 90 Minuten Dauer (0.7) Benotete Übungen - studienbegleitend (0.3)
18. Grundlage für ... :	
19. Medienform:	
20. Prüfungsnummer/n und -name:	<ul style="list-style-type: none">• 11111 Verteilte Systeme• 11112 Verteilte Systeme - Übungen
21. Angeboten von:	Institut für Parallele und Verteilte Systeme
22. Zuordnung zu weiteren Curricula:	<p>B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Wirtschaftsinformatik, 5. Semester → Informatik (B 1)</p> <p>B.Sc. Wirtschaftsinformatik, 5. Semester → Wahlpflichtbereich (Bereich C) → Wahlbereich II: Informatik Basis</p> <p>B.Sc. Wirtschaftsinformatik, 5. Semester → Wahlpflichtbereich (Bereich C) → Wahlbereich III: Informatik Simulation</p>

330 Katalog ISW

Zugeordnete Module:

- 10040 Bildsynthese
- 10050 Bildverstehen
- 10060 Computergraphik
- 11900 Design and Test of Systems on a Chip
- 14380 Hardware Verification and Quality Assessment
- 10160 Hardwarebeschreibungssprachen
- 10180 Information Retrieval und Text Mining
- 10300 Rechnernetze
- 11330 Visualisierung

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:	Thomas Ertl		
9. Dozenten:	<ul style="list-style-type: none"> • Thomas Ertl • Daniel Weiskopf 		
10. Zuordnung zum Curriculum:	<p>B.Sc. Informatik, 4. Semester</p> <ul style="list-style-type: none"> → Ergänzungsmodule → Wahlmodule aus Master Informatik <p>BA (Komb) Informatik, 4. Semester</p> <ul style="list-style-type: none"> → Module im Nebenfach → Katalog ISW 		
11. Voraussetzungen:	<ul style="list-style-type: none"> • Modul 051900002 Computergraphik • Modul 051900001 Mensch-Computer-Interaktion 		
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.</p> <p>Interaktive Verfahren nutzen spezielle Eigenschaften moderner Graphikhardware, um mit Hilfe mehrdimensionaler Texturen und anderer Rasterisierungsoperationen realistische Beleuchtungseffekte in Echtzeit zu generieren. 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 		
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 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 100401 Vorlesung Bildsynthese • 100402 Übung Bildsynthese 		

16. Abschaetzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17a. Studienleistung:	Prüfungsvorleistung: Übungsschein.
17b. Prüfungsleistungen:	Mündliche Prüfung von 30 Minuten Dauer.
18. Grundlage für ... :	
19. Medienform:	
20. Prüfungsnummer/n und -name:	10041 Bildsynthese
21. Angeboten von:	
22. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 6. Semester → Ergänzungsmodule → Katalog ISW

Modul: 10050 Bildverstehen

2. Modulkürzel:	051200035	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Paul Levi		
9. Dozenten:	<ul style="list-style-type: none"> • Paul Levi • Viktor Avrutin 		
10. Zuordnung zum Curriculum:	<p>B.Sc. Informatik, 4. Semester → Ergänzungsmodule → Katalog ISW 1-3</p> <p>BA (Komb) Informatik, 4. Semester → Module im Nebenfach → Katalog ISW</p>		
11. Voraussetzungen:	Grundwissen über Programmierung, Datenstrukturen und Mathematik.		
12. Lernziele:	<p>Die Teilnehmer kennen und verstehen die Grundlagen der klassischen, verkörperten und verteilten Künstlichen Intelligenz. Ihnen wurden die dabei verwendeten Grundbegriffe so vorgestellt, dass sie als Bausteine von Architekturen intelligenter Systeme betrachtet werden. Dies bedeutet im Einzelnen, dass die Zuhörer am Beispiel des Bildverstehens die Wirkungsweise kognitiver Wahrnehmungsfähigkeiten kennengelernt haben, verstehen wie diese Fähigkeiten im Rahmen des Beobachtungs-, Planungs- und Aktionszyklus in einem Agenten eingesetzt werden und wie diese Fähigkeiten erweitert werden müssen, damit einzelne Agenten sich in einem Team kooperativ (Multiagenten-Systeme) verhalten können.</p> <p>Die Studierenden sind in der Lage, nicht nur einzelne wesentliche Algorithmen der Künstlichen Intelligenz (einschließlich des Bildverstehens) zu verstehen, sondern auch die wechselseitigen Beziehungen von verschiedenen Methoden zu berücksichtigen, um eine Beurteilung der Konzeption und der Wirkungsweise von intelligenten (kognitiven) technischen Systemen selbst vornehmen zu können.</p>		
13. Inhalt:	<ul style="list-style-type: none"> • Einleitung in Problemstellungen und Definitionen • Einführung in die Grundlagen der Bildverarbeitung • (Künstliche) Neuronale Netze • Bedingungsausbreitung (Constraints und ihre Propagierung) • Probabilistische Inferenz mit Bayes-Netzwerke • Verteilte Künstliche Intelligenz (VKI) • Multiagentensysteme (MAS) 		
14. Literatur:	<ul style="list-style-type: none"> • G. F. Luger, Künstliche Intelligenz, 2001 • Jähne, Bernd, Digitale Bildverarbeitung, 2005 • Literatur, siehe Webseite zur Veranstaltung • S. Russell, P. Norvig, Künstliche Intelligenz, 2004 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 100501 Vorlesung Bildverstehen • 100502 Übung Bildverstehen 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17a. Studienleistung:	keine		

17b. Prüfungsleistungen: Schriftliche Prüfung von 90 Minuten Dauer.

18. Grundlage für ... :

19. Medienform:

20. Prüfungsnummer/n und -name: 10051 Bildverstehen

21. Angeboten von:

22. Zuordnung zu weiteren Curricula: B.Sc. Softwaretechnik, 4. Semester
→ Ergänzungsmodule
→ Katalog ISW

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:	Daniel Weiskopf		
9. Dozenten:	<ul style="list-style-type: none"> • Thomas Ertl • Daniel Weiskopf 		
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 4. Semester → Ergänzungsmodule → Katalog ISW 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISW		
11. 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:	Folgende Themen werden in der Vorlesung behandelt: <ul style="list-style-type: none"> • Überblick über den Bildsyntheseprozess • Grundlegende Rastergraphik • 2D und 3D Geometrietransformationen, 3D Projektion • Polygonale und hierarchische Modelle • Verdeckungsrechnung • Grundlegende Renderingtechniken (Rasterung, Raytracing) • Beleuchtungsmodelle • Texturen • Grundlagen der geometrischen Modellierung (Kurven, Flächen) Die Veranstaltung besteht aus Vorlesung mit Übungen. Die Übungen umfassen praktische Programmierübungen, theoretische Themen und Programmierprojekte.		
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. Abschaetzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17a. Studienleistung:	Prüfungsvorleistung: Übungsschein.		
17b. Prüfungsleistungen:	Schriftliche Prüfung von 120 Minuten Dauer.		
18. Grundlage für ... :			
19. Medienform:			

20. Prüfungsnummer/n und -name: 10061 Computergraphik

21. Angeboten von:

22. Zuordnung zu weiteren Curricula: B.Sc. Softwaretechnik, 5. Semester
→ Ergänzungsmodule
→ Katalog ISW
B.Sc. Maschinelle Sprachverarbeitung, 5. Semester
→ Wahlbereich E/I

Modul: 11900 Design and Test of Systems on a Chip

2. Modulkürzel:	051700015	5. Moduldauer:	-
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Hans-Joachim Wunderlich		
9. Dozenten:	<ul style="list-style-type: none"> • Hans-Joachim Wunderlich • Melanie Elm 		
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 4. Semester → Ergänzungsmodule → Katalog ISW 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISW		
11. Voraussetzungen:	<ul style="list-style-type: none"> • Modul 051700005 Rechnerorganisation • Modul 051700010 Grundlagen der Rechnerarchitektur 		
12. Lernziele:	<p>The students of this course have gained a basic understanding of development and test of complex embedded hardware / software systems. The participants have become acquainted with the essential steps of synthesis, validation, test and programming and have learned, how to use the related tools for automation.</p> <p>Besides the different design styles, paradigms and standards the essential steps of automated design, test and programming of digital and mixed signal circuits have been discussed. Exercises and labs have served to practice the use of commercial tools and designs.</p>		
13. Inhalt:	<p>The course comprises:</p> <ul style="list-style-type: none"> • Overview over system design • Reuse and cores • Standards and platforms • Elements of analog and mixed signal designs • Design validation and verification • Test and design for testability with the related standards • Application and programming of embedded processors 		
14. Literatur:	<ul style="list-style-type: none"> • A. Sloss, D. Symes, C. Wright, ARM System Developer's Guide: Designing and Optimizing System Software, 2004 • L.-T. Wang, C.-W. Wu, X. Wen, VLSI Test Principles and Architectures - Design for Testability, 2006 • M. Keating, P. Bricaud, Reuse Methodology Manual for System-on-a-Chip Designs, 2007 • M. L. Bushnell, V. D. Agrawal, Essentials of Electronic Testing, 2005 • S. Furber, ARM System-on-Chip Architecture, 2000 • W. Wolf, Modern VLSI Design: System-on-Chip Design, 2002 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 119001 Vorlesung Design and Test of Systems on a Chip • 119002 Übung Design and Test of Systems on a Chip • 119003 Praktikum Design and Test of Systems on a Chip 		
16. Abschaetzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		

17a. Studienleistung:	Keine
17b. Prüfungsleistungen:	Lehrveranstaltungsbegleitende Prüfung (0.30) Schriftliche Prüfung von 90 Minuten Dauer (0.70)
18. Grundlage für ... :	
19. Medienform:	
20. Prüfungsnummer/n und -name:	11901 Design and Test of Systems on a Chip
21. Angeboten von:	
22. Zuordnung zu weiteren Curricula:	

Modul: 14380 Hardware Verification and Quality Assessment

2. Modulkürzel:	051700020	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Hans-Joachim Wunderlich		
9. Dozenten:	<ul style="list-style-type: none"> • Hans-Joachim Wunderlich • Michael Kochte 		
10. Zuordnung zum Curriculum:	<p>B.Sc. Informatik, 4. Semester</p> <ul style="list-style-type: none"> → Ergänzungsmodule → Wahlmodule aus Master Informatik <p>BA (Komb) Informatik, 4. Semester</p> <ul style="list-style-type: none"> → Module im Nebenfach → Katalog ISW 		
11. Voraussetzungen:	<ul style="list-style-type: none"> • 051700005 Rechnerorganisation • 051711005 Technische Grundlagen der Informatik 		
12. Lernziele:	<ul style="list-style-type: none"> • Basic knowledge of methodologies and algorithms of functional and formal verification, diagnosis, test and design for testability of integrated circuits • Application of tools for simulation, verification and test insertion 		
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 • W. Kunz, D. Stoffel, Reasoning in Boolean Networks, 1997 		
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. Abschaetzung Arbeitsaufwand:	<p>Präsenzzeit: 42 Stunden</p> <p>Nachbearbeitungszeit: 138 Stunden</p>		

17a. Studienleistung:	Keine
17b. Prüfungsleistungen:	Schriftliche Prüfung von 90 Minuten Dauer
18. Grundlage für ... :	
19. Medienform:	
20. Prüfungsnummer/n und -name:	14381 Hardware Verification and Quality Assessment
21. Angeboten von:	
22. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW

Modul: 10160 Hardwarebeschreibungssprachen

2. Modulkürzel:	051711015	5. Moduldauer:	1 Semester
3. Leistungspunkte:	3.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	2.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Martin Radetzki		
9. Dozenten:	<ul style="list-style-type: none"> • Martin Radetzki • Sven Simon 		
10. Zuordnung zum Curriculum:	<p>B.Sc. Informatik, 4. Semester</p> <ul style="list-style-type: none"> → Ergänzungsmodule → Katalog ISW 4-7 <p>BA (Komb) Informatik, 4. Semester</p> <ul style="list-style-type: none"> → Module im Nebenfach → Katalog ISW 		
11. Voraussetzungen:	<ul style="list-style-type: none"> • Modul 051711005 Technische Grundlagen der Informatik <p>oder</p> <ul style="list-style-type: none"> • Modul Einführung in die Technische Informatik (aus Studiengang Softwaretechnik, Modulnummer wird nachgetragen) 		
12. Lernziele:	<ul style="list-style-type: none"> • Kenntnis Hardwarebeschreibungssprachen • Kenntnis VHDL 		
13. Inhalt:	<p>Hardwarebeschreibungssprachen werden eingesetzt, um digitale Schaltungen auf der Algorithmen-, Registertransfer- und Gatterebene zu beschreiben. Sie ermöglichen eine ereignisgesteuerte Simulation und stellen Konzepte zur Beschreibung von Hierarchie, Nebenläufigkeit und Zeit zusätzlich zu den von Programmiersprachen bekannten Eigenschaften zur Verfügung. Dieses Modul vermittelt die wichtigsten Sprachkonstrukte der Hardwarebeschreibungssprache VHDL und ihre sinnvolle Anwendung zur Beschreibung digitaler Schaltungen und Systeme, indem die folgenden Themen vorgestellt und in Rechnerübungen angewandt werden:</p> <ul style="list-style-type: none"> • Entwurfshierarchie: Entities, Architectures, Instanzen, Verbindungen • Quellcode Dateien und ihre Kompilation in Entwurfsbibliotheken • Nebenläufige sequentielle Prozesse • Sequentielle Anweisungen in VHDL • Typsystem von VHDL • Beschreibung typischer Hardwarestrukturen • VHDL für die Hardware synthese, Synthesese mantik • Beschreibung regulärer und rekursiver Strukturen • Simulationsmechanismen • Testbenches and Handhabung von Dateien • Organisation VHDL-basierter Projekte 		
14. Literatur:	<ul style="list-style-type: none"> • P. J. Ashenden, The Designer's Guide to VHDL (2nd edition), 2002 • P. J. Ashenden, The Student's Guide to VHDL, 1998 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 101601 Vorlesung Hardwarebeschreibungssprachen • 101602 Übung Hardwarebeschreibungssprachen 		
16. Abschaetzung Arbeitsaufwand:	Präsenzzeit: 21 Stunden		

	Nachbearbeitungszeit: 69 Stunden
17a. Studienleistung:	Keine
17b. Prüfungsleistungen:	Bewertete Rechnerübungen - vorlesungsbegleitend (0.5) Schriftliche Prüfung von 60 Minuten Dauer (0.5)
18. Grundlage für ... :	
19. Medienform:	
20. Prüfungsnummer/n und -name:	<ul style="list-style-type: none">• 10161 Hardwarebeschreibungssprachen• 10162 Hardwarebeschreibungssprachen - Rechnerübungen
21. Angeboten von:	
22. Zuordnung zu weiteren Curricula:	

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:	Hinrich Schütze		
9. Dozenten:	<ul style="list-style-type: none"> • Helmut Schmid • Hinrich Schütze 		
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 4. Semester → Ergänzungsmodule → Katalog ISW 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISW		
11. 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. Abschaetzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17a. Studienleistung:	regelmäßige Hausübungen		
17b. Prüfungsleistungen:	Klausur, Gewicht 1,0, schriftlich, 60 Minuten		
18. Grundlage für ... :			
19. Medienform:			
20. Prüfungsnummer/n und -name:	10181 Information Retrieval und Text Mining		
21. Angeboten von:			
22. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISW B.Sc. Maschinelle Sprachverarbeitung, 5. Semester → Kernmodule		

Modul: 10300 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:		Kurt Rothermel	
9. Dozenten:		Kurt Rothermel	
10. Zuordnung zum Curriculum:		B.Sc. Informatik, 4. Semester → Ergänzungsmodule → Katalog ISW 1-3 BA (Komb) Informatik, 4. Semester → Module im Nebenfach → Katalog ISW	
11. Voraussetzungen:		<ul style="list-style-type: none"> • 051520005 Programmierung und Software-Entwicklung • 051510005 Datenstrukturen und Algorithmen 	
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"> • 103001 VL Rechnernetze • 103002 ÜB Rechnernetze 	
16. Abschaetzung Arbeitsaufwand:		Präsenzzeit: 42 Stunden	

Nachbearbeitungszeit: 138 Stunden

17a. Studienleistung: Keine

17b. Prüfungsleistungen: Schriftliche Prüfung von 90 Minuten Dauer (0.7)
Benotete Übungen - studienbegleitend (0.3)

18. Grundlage für ... :

19. Medienform:

20. Prüfungsnummer/n und -name:

- 10301 Rechnernetze
- 10302 Rechnernetze - Übungen

21. Angeboten von:

22. Zuordnung zu weiteren Curricula: B.Sc. Softwaretechnik, 4. Semester
→ Ergänzungsmodule
→ Katalog ISW

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:	Deutsch
8. Modulverantwortlicher:	Daniel Weiskopf		
9. Dozenten:	<ul style="list-style-type: none"> • Thomas Ertl • Daniel Weiskopf 		
10. Zuordnung zum Curriculum:	<p>B.Sc. Informatik, 4. Semester</p> <ul style="list-style-type: none"> → Ergänzungsmodule → Wahlmodule aus Master Informatik <p>BA (Komb) Informatik, 6. Semester</p> <ul style="list-style-type: none"> → Module im Nebenfach → Katalog ISW 		
11. Voraussetzungen:	<ul style="list-style-type: none"> • 051900002 Computergraphik • 051900001 Mensch-Computer-Interaktion • 051240005 Numerik und Stochastik. 		
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 • Hochdimensionale Daten und Informationsvisualisierung 		
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 • 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. Abschaetzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17a. Studienleistung:	Prüfungsvorleistung: Übungsschein.
17b. Prüfungsleistungen:	Mündliche Prüfung von 30 Minuten Dauer.
18. Grundlage für ... :	
19. Medienform:	
20. Prüfungsnummer/n und -name:	11331 Visualisierung
21. Angeboten von:	
22. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 6. Semester → Ergänzungsmodule → Katalog ISW B.Sc. Maschinelle Sprachverarbeitung, 5. Semester → Wahlbereich E/I

Modul: 10260 Programmierkurs

2. Modulkürzel:	051520010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	3.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	2.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Jochen Ludewig		
9. Dozenten:	<ul style="list-style-type: none"> • • Sascha Riexinger • Holger Röder 		
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 1. Semester → Basismodule BA (Komb) Informatik, 1. Semester → Module im Nebenfach		
11. Voraussetzungen:	Keine		
12. Lernziele:	Selbstständiges Erstellen von Programmen und Lösung von Programmieraufgaben in einer vorgegebenen Programmiersprache wie Java.		
13. Inhalt:	<p>Der Programmierkurs ergänzt die Vorlesung Programmierung und Software-Entwicklung (PSE). Die Teilnehmer erlernen eine weitere Programmiersprache (Java). Ihre Merkmale, Syntax und Semantik, werden denen der in PSE gelehrt Sprache gegenübergestellt. Praktische Übungen bereiten die Teilnehmer auf die Bearbeitung der Schein-Aufgabe vor.</p> <p>Die Lehrveranstaltung findet in zwei Varianten statt. Die Teilnahme richtet sich nach dem Studiengang:</p> <p>S. Riexinger:</p> <ul style="list-style-type: none"> • BSc. Informatik • BA (Komb) Informatik • BSc. Maschinelle Sprachverarbeitung <p>H. Röder:</p> <ul style="list-style-type: none"> • BSc. Softwaretechnik • BSc. Wirtschaftsinformatik • BSc. Technikpädagogik • MSc. Technikpädagogik 		
14. Literatur:			
15. Lehrveranstaltungen und -formen:	102601 Übung Programmierkurs		
16. Abschaetzung Arbeitsaufwand:	Präsenzzeit: 21 Stunden Nachbearbeitungszeit: 69 Stunden		
17a. Studienleistung:	USL (Übungsschein - Scheinkriterien werden zu Beginn der Veranstaltung angekündigt.)		
17b. Prüfungsleistungen:	-		
18. Grundlage für ... :			

19. Medienform:

20. Prüfungsnummer/n und -name: 10261 Programmierkurs

21. Angeboten von:

22. Zuordnung zu weiteren Curricula:

- B.Sc. Softwaretechnik, 1. Semester
 - Basismodule
- B.Sc. Wirtschaftsinformatik, 1. Semester
 - Informatik (B 1)
- B.Sc. Maschinelle Sprachverarbeitung, 1. Semester
 - Basismodule
- B.Sc. Technikpädagogik, 1. Semester
 - Hauptfach Informatik
 - Basismodule Informatik
- B.Sc. Technikpädagogik, 1. Semester
 - Wahlpflichtfach
 - Informatik
- M.Sc. Technikpädagogik, 1. Semester
 - Wahlpflichtfach B
 - Affines Wahlpflichtfach Informatik
 - Affines Wahlpflichtfach Informatik Basismodule

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:	Bernhard Mitschang		
9. Dozenten:	Bernhard Mitschang		
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 1. Semester → Basismodule BA (Komb) Informatik, 1. Semester → Module im Nebenfach		
11. 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"> • Algorithmen und funktionale Programmierung Kap. 1 verwendet nur die funktionale Teilmenge der Programmiersprache Ada, keine Variablen, keine Prozeduren. Grammatik, Formale Sprachen und BNF werden eingeführt. • Imperative Programmierung Kap. 2 erweitert die verwendete Sprache durch die prozeduralen Konzepte, also Variablen und Prozeduren. Zu den Sprachkonstrukten werden Vor- und Nachbedingungen, mit den Schleifen die Invarianten eingeführt. Datentypen werden schrittweise ausgebaut. In Zusammenhang mit den Zeigern werden die Konzepte für Keller und Halde vermittelt. Die Entwicklung einfacher Programme wird gezeigt und geübt. • Aufbau und Organisation komplexer Programme. Die Modularisierung, die bei größeren Programmen notwendig ist, führt zur Kapselung und zu den abstrakten Datentypen. Damit entsteht die Möglichkeit, neue Datenstrukturen und Datentypen sicher zu definieren. Die Konzepte der Kompilation und der Interpretation werden erläutert. Wichtige Beispiele komplexer Datentypen werden entwickelt. Die Konzepte der Generalisierung (generische Einheiten) werden vermittelt. • Ausnahmebehandlung Möglichkeiten und Probleme der Ausnahmebehandlung sind Gegenstände dieses kurzen Kapitels. • Objektorientierte Programmierung Am Ende des Semesters steht ein Ausblick in die objektorientierte Programmierung, d.h. die Umsetzung der bereits bekannten Konzepte (ADTs) in die objektorientierte Sichtweise und die Vererbung. Dieser Teil bereitet die Programmierung in einer objektorientierten Sprache (3. Semester) vor. 		
14. Literatur:	Manuskripte: V.Claus (WS 08/09 bis SS 2009)		

Appelrath, Hans-Jürgen und Ludewig, Jochen, "Skriptum Informatik - eine konventionelle Einführung", Verlag der Fachvereine Zürich und B.G. Teubner Stuttgart, 4. Auflage 1999

Nagl., M., "Softwaretechnik mit Ada 95. Entwicklung großer Systeme.", Vieweg-Verlag, Wiesbaden 1999

Barnes, J.G.P., "Programming in Ada 95", 2. Auflage, Addison-Wesley 1998

15. Lehrveranstaltungen und -formen: • 102801 Vorlesung Programmierung und Softwareentwicklung
 • 102802 Übung Programmierung und Softwareentwicklung

16. Abschaetzung Arbeitsaufwand: Präsenzzeit: 63 Stunden
 Vor-/Nachbearbeitungszeit: 187 Stunden
 Prüfungsvorbereitung: 20 Stunden

17a. Studienleistung: Studienleistung: Übungsschein, Vor. 3 mal vortragen in den
 Übungen und mindestens 50% der Übungspunkte erwerben,
 Teilnahme an den Zwischenklausuren.

17b. Prüfungsleistungen: Modulprüfung: Schriftliche Prüfung von 120 Minuten Dauer

18. Grundlage für ... :

19. Medienform:

20. Prüfungsnummer/n und -name: 10281 Programmierung und Software-Entwicklung

21. Angeboten von:

22. Zuordnung zu weiteren Curricula: B.Sc. Softwaretechnik, 1. Semester
 → Basismodule
 B.Sc. Wirtschaftsinformatik, 1. Semester
 → Informatik (B 1)
 B.Sc. Mechatronik, 1. Semester
 → Kernmodule
 B.Sc. Maschinelle Sprachverarbeitung, 1. Semester
 → Basismodule
 B.Sc. Technikpädagogik, 1. Semester
 → Hauptfach Informatik
 → Basismodule Informatik
 B.Sc. Technikpädagogik, 1. Semester
 → Wahlpflichtfach
 → Informatik
 M.Sc. Technikpädagogik, 1. Semester
 → Wahlpflichtfach B
 → Affines Wahlpflichtfach Informatik
 → Affines Wahlpflichtfach Informatik Basismodule
 B.Sc. Simulation Technology, 1. Semester
 → Grundstudium
 ohne Absch Lehramt-Pool, 1. Semester
 → Pflichtmodule
