

Modulhandbuch
Studiengang Bachelor of Arts (Lehramt) Informatik HF
Prüfungsordnung: 079-1-2015

Sommersemester 2017
Stand: 03. April 2017

Universität Stuttgart
Keplerstr. 7
70174 Stuttgart

Kontaktpersonen:

Studiengangsmanager/in:

Katrin Schneider
Institut für Maschinelle Sprachverarbeitung
Tel.: 685 88520
E-Mail: katrin.schneider@informatik.uni-stuttgart.de

Inhaltsverzeichnis

100 Pflichtmodule	4
10280 Programmierung und Software-Entwicklung	5
10940 Theoretische Grundlagen der Informatik	7
110 weitere Pflichtmodule	9
1110 zweites Hauptfach Mathematik	10
11890 Algorithmen und Berechenbarkeit	11
1120 zweites Hauptfach nicht Mathematik	12
10190 Mathematik für Informatiker und Softwaretechniker	13
12060 Datenstrukturen und Algorithmen	15
14360 Einführung in die Technische Informatik	17
200 Fachdidaktik	19
34060 Grundlagen der Fachdidaktik Informatik	20
300 Wahlmodule	22
310 Auswahl 1	23
48510 Key Qualifications in Tutoring Computer Science	24
58720 Seminar-LA-INF	25
60340 Seminar-INF 2	27
320 Auswahl 2	29
10210 Mensch-Computer-Interaktion	30
10220 Modellierung	32
36100 Programmierparadigmen	34
40090 Systemkonzepte und -programmierung	36

100 Pflichtmodule

Zugeordnete Module: 10280 Programmierung und Software-Entwicklung
 10940 Theoretische Grundlagen der Informatik
 110 weitere Pflichtmodule
 12060 Datenstrukturen und Algorithmen
 14360 Einführung in die Technische Informatik

Modul: 10280 Programmierung und Software-Entwicklung

2. Modulkürzel:	051520005	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	9 LP	6. Turnus:	Wintersemester
4. SWS:	6	7. Sprache:	Deutsch
8. Modulverantwortlicher:		Frank Leymann	
9. Dozenten:		Frank Leymann	
10. Zuordnung zum Curriculum in diesem Studiengang:		B.A. (L) Informatik HF, PO 079-1-2015, 1. Semester → Pflichtmodule	
11. Empfohlene Voraussetzungen:		Keine. Teilnahme an einem Vorkurs Java ist hilfreich aber nicht notwendig.	
12. Lernziele:		Die Teilnehmer haben einen Überblick über das Gebiet der Informatik. Sie haben die wichtigsten Konzepte einer höheren Programmiersprache und ihrer Verwendung verstanden und sind in der Lage, kleine Programme (bis zu einigen hundert Zeilen) zu analysieren und selbst zu konzipieren und zu implementieren. Sie kennen die Möglichkeiten, Daten- und Ablaufstrukturen zu entwerfen, zu beschreiben und zu codieren. Sie haben die Abstraktionskonzepte moderner Programmiersprachen verstanden. Sie kennen die Techniken und Notationen zur Definition kontextfreier Programmiersprachen und können damit arbeiten.	
13. Inhalt:		<ul style="list-style-type: none"> - 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:		Präsenzstunden: 63 h Eigenstudiumstunden: 207 h Gesamtstunden: 270 h	

17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none">• 10281 Programmierung und Software-Entwicklung (PL), Schriftlich, 90 Min., Gewichtung: 1• V Vorleistung (USL-V), Schriftlich oder Mündlich [10281] Programmierung und Software-Entwicklung (PL), schriftliche Prüfung, 90 Min., Gewicht: 1.0, [Prüfungsvorleistung] Vorleistung (USL-V), schriftlich, eventuell mündlich, Vorleistung: Übungsschein. Voraussetzungen werden zu Beginn vom Dozenten festgesetzt. Dazu gehören eine bestimmte Anzahl von Vorträgen in den Übungen und ein bestimmter Teil der Übungspunkte.
18. Grundlage für ... :	Datenstrukturen und Algorithmen
19. Medienform:	<ul style="list-style-type: none">- Folien über Beamer- Tafelanschrieb
20. Angeboten von:	Architektur von Anwendungssystemen

Modul: 10940 Theoretische Grundlagen der Informatik

2. Modulkürzel:	050420005	5. Moduldauer:	Zweimestrig
3. Leistungspunkte:	12 LP	6. Turnus:	Wintersemester
4. SWS:	8	7. Sprache:	Deutsch
8. Modulverantwortlicher:	apl. Prof. Dr. rer. nat. habil. Ulrich Hertrampf		
9. Dozenten:	Volker Diekert Ulrich Hertrampf		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.A. (L) Informatik HF, PO 079-1-2015, 1. Semester → Pflichtmodule		
11. Empfohlene Voraussetzungen:	Keine		
12. Lernziele:	<ul style="list-style-type: none"> • Logik und Diskrete Strukturen: 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. • Automaten und Formale Sprachen: 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. 		
13. Inhalt:	<p>Logik und Diskrete Strukturen:</p> <ul style="list-style-type: none"> • Einführung in die Aussagenlogik: Semantik (Wahrheitswerte), Syntax (Axiome und Schlussregeln), Normalformen, Hornformeln, Endlichkeitssatz, aussagenlogische Resolution, • Einführung in die Prädikatenlogik 1. Stufe: Semantik und Syntax, Normalformen, Unifikatoren, Herbrand-Theorie, prädikatenlogische Resolution, • Elementare Zahlentheorie: Rechnen mit Restklassen, endliche Körper, Euklidischer Algorithmus, Chinesischer Restsatz, Primzahltests, RSA-Verfahren, Wachstumsabschätzungen, Grundbegriffe der Wahrscheinlichkeitsrechnung, Kombinatorik, Graphen. <p>Automaten und Formale Sprachen:</p> <ul style="list-style-type: none"> • 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. 		
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 • 109403 Vorlesung Automaten und Formale Sprachen • 109404 Übung Automaten und Formale Sprachen • 109402 Übung Logik und Diskrete Strukturen 		

- 109405 Zusatztutorial Theoretische Grundlagen der Informatik für MSV (freiwillig)
-

16. Abschätzung Arbeitsaufwand:

17. Prüfungsnummer/n und -name:

- 10941 Theoretische Grundlagen der Informatik (PL), Schriftlich, 120 Min., Gewichtung: 1
 - V Vorleistung (USL-V), Schriftlich oder Mündlich, 30 Min.
- [10941] Theoretische Grundlagen der Informatik (PL), schriftliche Prüfung, 120 Min., Gewicht: 1.0, Prüfungsvorleistung: Übungsschein [Prüfungsvorleistung] Vorleistung (USL-V), schriftlich, eventuell mündlich, 30 Min.
-

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

Theoretische Informatik

110 weitere Pflichtmodule

Zugeordnete Module:	1110	zweites Hauptfach Mathematik
	1120	zweites Hauptfach nicht Mathematik

1110 zweites Hauptfach Mathematik

Zugeordnete Module: 11890 Algorithmen und Berechenbarkeit

Modul: 11890 Algorithmen und Berechenbarkeit

2. Modulkürzel:	050420020	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Dr. Stefan Funke		
9. Dozenten:	Ulrich Hertrampf Volker Diekert Stefan Funke		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.A. (L) Informatik HF, PO 079-1-2015, 3. Semester → zweites Hauptfach Mathematik --> weitere Pflichtmodule --> Pflichtmodule		
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	
	Selbststudiums- /	138 h	
	Nacharbeitszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 11891 Algorithmen und Berechenbarkeit (PL), Schriftlich, 120 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich, 30 Min. Prüfungsvorleistung: Übungsschein 		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Algorithmik		

1120 zweites Hauptfach nicht Mathematik

Zugeordnete Module: 10190 Mathematik für Informatiker und Softwaretechniker

Modul: 10190 Mathematik für Informatiker und Softwaretechniker

2. Modulkürzel:	080300100	5. Moduldauer:	Zweisemestrig
3. Leistungspunkte:	18 LP	6. Turnus:	Wintersemester
4. SWS:	12	7. Sprache:	Deutsch
8. Modulverantwortlicher:	PD Dr. Andreas Markus Kollross		
9. Dozenten:	Wolfgang Rump Andreas Markus Kollross Peter Lesky Wolf-Patrick Düll		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.A. (L) Informatik HF, PO 079-1-2015, 3. Semester → zweites Hauptfach nicht Mathematik --> weitere Pflichtmodule --> Pflichtmodule		
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:	<p>1. Semester:</p> <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) <p>2. Semester:</p> <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, Vorlesungsskripte , 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), Schriftlich, 120 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich 		

Ein Übungsschein aus den beiden Veranstaltungen, jeweils im 1. oder 2. Fachsemester zu erwerben

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Geometrie

Modul: 12060 Datenstrukturen und Algorithmen

2. Modulkürzel:	051510005	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	9 LP	6. Turnus:	Sommersemester
4. SWS:	6	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Dr. Daniel Weiskopf		
9. Dozenten:	Stefan Funke Thomas Ertl Andrés Bruhn Daniel Weiskopf		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.A. (L) Informatik HF, PO 079-1-2015, 2. Semester → Pflichtmodule		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • Modul 10280 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>Die Lernziele lassen sich wie folgt zusammenfassen:</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 		
13. Inhalt:	<p>Es werden die folgenden Themen behandelt:</p> <ul style="list-style-type: none"> • Vorgehensweise bei der Entwicklung und Implementierung von Algorithmen • Komplexität und Effizienz von Algorithmen, O-Notation • Listen (Stack, Queue, doppelt verkettete Listen) • Sortierverfahren (Selection-, Insertion-, Bubble-, Merge-, Quick-Sort) • Bäume (Binär-, AVL-, 2-3-4-, Rot-Schwarz-, B-Bäume, Suchbäume, Traversierung, Heap) • Räumliche Datenstrukturen (uniforme Gitter, Oktal-, BSP-, kD-, CSG-Bäume, Bounding-Volumes) • Graphen (Datenstrukturen, DFS, BFS, topologische Traversierung, Dijkstra-, A*, Bellman-Ford-Algorithmen, minimale Spannbäume, maximaler Fluss) • Räumliche Graphen (Triangulierung, Voronoi, Delaunay, Graph-Layout) • Textalgorithmen (String-Matching, Knuth-Morris-Pratt, Boyer-Moore, reguläre Ausdrücke, Levenshtein-Distanz) • Hashing (Hashfunktionen, Kollisionen) • Verteilte Algorithmen (Petri-Netze, Programmieren nebenläufiger Abläufe, einige parallele und parallelisierte Algorithmen) 		

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

14. Literatur:

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

15. Lehrveranstaltungen und -formen:

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

16. Abschätzung Arbeitsaufwand:

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

17. Prüfungsnummer/n und -name:

- 12061 Datenstrukturen und Algorithmen (PL), Schriftlich, 120 Min., Gewichtung: 1
 - V Vorleistung (USL-V), Schriftlich oder Mündlich
- Prüfungsvorleistung: Übungsschein.
Die genauen Details der Übungsleistungen und Ihrer Anrechnung werden zu Beginn der Lehrveranstaltung bekannt gegeben.

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

Visualisierung

Modul: 14360 Einführung in die Technische Informatik

2. Modulkürzel:	051400105	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Dr. Otto Eggenberger		
9. Dozenten:	Otto Eggenberger Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.A. (L) Informatik HF, PO 079-1-2015, 5. Semester → Pflichtmodule		
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:	<p>Grundlegende Funktionsweise eines Computers</p> <ul style="list-style-type: none"> - Informationsdarstellung - Zahlendarstellung und Codes - Digitale Grundbausteine - Logische Funktionen, Speicherelemente - Befehlsausführung, Programmablauf Elektrotechnische Grundlagen - Physikalische Grundbegriffe - Elektrische Spannung, elektrischer Strom - Elektrische Netzwerke - Halbleiterbauelemente - Digitale Grundsaltungen Digitale Schaltungen - Schaltnetzwerke - Boolesche Algebra und Schaltalgebra - Darstellung und Minimierung von Schaltfunktionen - Rückkopplung, Zustandsbegriff - Automaten und sequentielle Netzwerke - Digitale Standardschaltungen - Entwurfsmethodik 		
14. Literatur:	<ul style="list-style-type: none"> - 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"> • 143602 Gruppenübungen Einführung in die Technische Informatik • 143601 Vorlesung Einführung in die Technische Informatik 		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzstunden: 42 h Eigenstudiumstunden: 138 h Gesamtstunden: 180 h</p>		
17. Prüfungsnummer/n und -name:	<p>14361 Einführung in die Technische Informatik (PL), Schriftlich, 60 Min., Gewichtung: 1 [14361] Einführung in die Technische Informatik (PL), schriftliche Prüfung, 60 Min., Gewicht: 1.0</p>		

18. Grundlage für ... : Rechnerorganisation 1

19. Medienform:

20. Angeboten von: Grundlagen der Informatik

200 Fachdidaktik

Zugeordnete Module: 34060 Grundlagen der Fachdidaktik Informatik

Modul: 34060 Grundlagen der Fachdidaktik Informatik

2. Modulkürzel:	101010062	5. Moduldauer:	Zweisemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Dr. Bernd Zinn		
9. Dozenten:	Andreas Mußotter Bernd Zinn		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.A. (L) Informatik HF, PO 079-1-2015, 6. Semester → Fachdidaktik		
11. Empfohlene Voraussetzungen:	Es wird empfohlen, die Fachdidaktik möglichst spät zu belegen.		
12. Lernziele:	<p><i>Die Studierenden sind in der Lage,</i></p> <p><i>die Fachdidaktik im Kontext der korrespondierenden Bezugsdisziplinen zu verorten und ihr Bedeutungsspektrum zu überblicken</i></p> <p><i>beruflich-technischen Unterricht zielorientiert zu planen und dabei didaktisch-methodische Bezugspunkte kriterienorientiert zu berücksichtigen</i></p> <p><i>beruflich-technische Konzepte des Unterrichts so zu gestalten, dass neben fachlich-methodischen auch sozial-kommunikative und personale Kompetenzen unter Berücksichtigung zentraler Aspekte (Umgang mit Inklusion und Heterogenität, Einsatz diagnostischer Verfahren) vermittelt werden können</i></p> <p><i>Erkenntnisse aus der (fachdidaktischen) Lehr-Lernforschung im Hinblick auf ihre Bedeutung für das Lehren und Lernen zu interpretieren und diese bei der Konzeptionierung von informationstechnischem Unterricht zu berücksichtigen</i></p> <p><i>den komplexen Prozess der Unterrichtsplanung, -durchführung und -evaluation von technischem Unterricht zu erfassen</i></p> <p><i>die Durchführung und Evaluation des Unterrichts in ihrer Komplexität als vielfältig interaktiven, inhaltsorientierten und insgesamt kriterienorientierten Prozess zu erfassen und die Ergebnisse kritisch zu reflektieren</i></p>		
13. Inhalt:	<p><i>Im Mittelpunkt des Moduls stehen folgende Lerninhalte:</i></p> <ul style="list-style-type: none"> <i>• Ausgangslage und Grundkonzeptionen der allgemeinen und beruflichen Technikdidaktik, Stellung der Fachdidaktik im Gefüge der Fachwissenschaft und Erziehungswissenschaft, zentrale Ansätze und Konzepte der beruflichen Bildung</i> <i>• methodisch-didaktische Ansätze im technischen Unterricht, Berufsfeldspezifische Aspekte (z.B. Lernen in technischen Reallernräumen, Experimente)</i> <i>• Umgang mit Inklusion und Heterogenität,</i> <i>• Pädagogische Diagnostik</i> 		

- *Analyse berufs- und schulformbezogener Lehrpläne*
 - *Planung, Durchführung und Evaluation von technischem Unterricht in der Aus- und Weiterbildung*
 - *Wandel beruflicher Anforderungen und Rahmenbedingungen in der Informatik*
 - *Aktuelle Inhalte der Lehr-Lernforschung im Bezugsfeld der Technikdidaktik und speziell Fachdidaktik Informatik*
-

14. Literatur:

- *Schubert, S. und Schwill, A. (2011): Didaktik der Informatik. 2. Aufl. Spektrum Akademischer Verlag.*
- *Tenberg, R. (2011): Vermittlung von fachlichen und überfachlichen Kompetenzen in technischen Berufen. Theorie und Praxis der Technikdidaktik. Stuttgart: Steiner.*
- *Nickolaus, R. (2008): Didaktik - Modelle und Konzepte beruflicher Bildung. Orientierungsleistungen für die Praxis. Baltmannsweiler: Schneider Hohengehren.*
- *Nickolaus, R. und Schanz, H. (Hrsg.)(2008): Didaktik der gewerblich-technischen Berufsbildung. In: Diskussion Berufsbildung, Bd. 9. Baltmannsweiler: Schneider Hohengehren.*
- *Kuhlmeier, W. (2005): Berufliche Fachdidaktiken zwischen Anspruch und Realität: Situationsanalyse und Perspektiven einer konzeptionellen Weiterentwicklung am Beispiel der Bereichsdidaktik Bau-, Holz- und Gestaltungstechnik. Bd. 3. Baltmannsweiler: Schneider Hohengehren.*

Aktuelle wissenschaftliche Zeitschriftenbeiträge, insbesondere aus der Lehr-Lernforschung, im Bezugsfeld der beruflichen Technikdidaktik.

15. Lehrveranstaltungen und -formen:

- 340601 Vorlesung Einführung in die Didaktik der Informatik
 - 340602 Seminar Vertiefungen zur Einführung in die Didaktik der Informatik
-

16. Abschätzung Arbeitsaufwand:

- *Präsenzzeit: 2 x 28 h = 56h*
- *Selbststudium: ca. 70 h (Vorlesung)*
- *Selbststudium: ca. 54 h (Seminar)*

Gesamt: ca. 180 h

17. Prüfungsnummer/n und -name:

- 34061 Grundlagen der Fachdidaktik Informatik (PL), Schriftlich oder Mündlich, Gewichtung: 1
 - 34062 Grundlagen der Fachdidaktik Informatik, Ausarbeitung inkl. Präsentation (USL), Schriftlich oder Mündlich, Gewichtung: 1
-

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

Berufspädagogik mit Schwerpunkt Technikdidaktik

300 Wahlmodule

Zugeordnete Module:	310	Auswahl 1
	320	Auswahl 2

310 Auswahl 1

Zugeordnete Module: 48510 Key Qualifications in Tutoring Computer Science
 58720 Seminar-LA-INF
 60340 Seminar-INF 2

Modul: 48510 Key Qualifications in Tutoring Computer Science

2. Modulkürzel:	-	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	3 LP	6. Turnus:	Wintersemester/ Sommersemester
4. SWS:	2	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Dr. Daniel Weiskopf		
9. Dozenten:	Dozenten der Informatik		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.A. (L) Informatik HF, PO 079-1-2015, 3. Semester → Auswahl 1 --> Wahlmodule		
11. Empfohlene Voraussetzungen:			
12. Lernziele:	Students learn theoretical and practical tutoring skills, especially in the context of teaching computer science		
13. Inhalt:	The course starts with an introduction to generic skills and key qualifications needed for tutoring exercise groups and labs in computer science. In particular, oral presentation skills, communication with other students, time management, and exercise grading skills are covered. This introduction is taught as a block course in the beginning of the semester. Then, the students will practice their tutoring skills by working as tutors in one of the undergraduate courses with computer science topics. Regular meetings with the course teachers are used to guide the students as tutors, provide feedback, and improve their tutoring skill.		
14. Literatur:	Will be given during the course.		
15. Lehrveranstaltungen und -formen:	• 485101 Course Key Qualifications in Tutoring Computer Science		
16. Abschätzung Arbeitsaufwand:	Präsenzstunden: 21 h Eigenstudiumstunden: 69 h Gesamtstunden: 90 h		
17. Prüfungsnummer/n und -name:	48511 Key Qualifications in Tutoring Computer Science (USL), Schriftlich oder Mündlich, Gewichtung: 1 [48511] Key Qualifications in Tutoring Computer Science (USL), schriftlich oder mündlich		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Visualisierung		

Modul: 58720 Seminar-LA-INF

2. Modulkürzel:	-	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	3 LP	6. Turnus:	Wintersemester/ Sommersemester
4. SWS:	2	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Dr. Stefan Wagner		
9. Dozenten:	Dozenten der Informatik		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.A. (L) Informatik HF, PO 079-1-2015, 4. Semester → Auswahl 1 --> Wahlmodule		
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.		
13. Inhalt:	Variabel: Es werden Seminare zu diversen, häufig aktuellen Themen angeboten. Das Seminar INF kann in der Informatik oder in einem affinen Fach durchgeführt werden, wie etwa Computerlinguistik, Elektrotechnik, Mathematik oder Wirtschaftswissenschaften. 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:	• 587201 Seminar LA-INF		
16. Abschätzung Arbeitsaufwand:	Präsenzstunden: 21 h Eigenstudiumstunden: 69 h Gesamtstunden: 90 h		
17. Prüfungsnummer/n und -name:	58721 Seminar-LA-INF (USL), Sonstige, Gewichtung: 1 [58721] Seminar-LA-INF (USL), schriftlich oder mündlich		

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Software Engineering

Modul: 60340 Seminar-INF 2

2. Modulkürzel:	050420095	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	3 LP	6. Turnus:	Wintersemester/ Sommersemester
4. SWS:	2	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Dr. Stefan Wagner		
9. Dozenten:			
10. Zuordnung zum Curriculum in diesem Studiengang:	B.A. (L) Informatik HF, PO 079-1-2015, 4. Semester → Auswahl 1 --> Wahlmodule		
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.		
13. Inhalt:	Variabel: Es werden Seminare zu diversen häufig aktuellen Themen angeboten. Das Seminar INF kann in der Informatik oder in einem affinen Fach durchgeführt werden, wie etwa Computerlinguistik, Elektrotechnik, Mathematik oder Wirtschaftswissenschaften. 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:	• 603401 Seminar		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 21 Stunden Nachbearbeitungszeit: 69 Stunden		
17. Prüfungsnummer/n und -name:	60341 Seminar-INF 2 (USL), Sonstige, Gewichtung: 1		
18. Grundlage für ... :			
19. Medienform:			

20. Angeboten von: Theoretische Informatik

320 Auswahl 2

Zugeordnete Module: 10210 Mensch-Computer-Interaktion
 10220 Modellierung
 36100 Programmierparadigmen
 40090 Systemkonzepte und -programmierung

Modul: 10210 Mensch-Computer-Interaktion

2. Modulkürzel:	051900001	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Dr. Albrecht Schmidt		
9. Dozenten:	Thomas Ertl Daniel Weiskopf Albrecht Schmidt Niels Henze		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.A. (L) Informatik HF, PO 079-1-2015, 5. Semester → Auswahl 2 --> Wahlmodule		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • Modul 10280 Programmierung und Software-Entwicklung 		
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. 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:	<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 h
	Selbststudiums- /	138 h
	Nachbearbeitungszeit:	
	Summe:	180 h
<hr/>		
17. Prüfungsnummer/n und -name:	• 10211 Mensch-Computer-Interaktion (PL), Schriftlich, 90 Min., Gewichtung: 1	
	• V Vorleistung (USL-V), Schriftlich oder Mündlich	
	Prüfungsvorleistung: Übungsschein	
<hr/>		
18. Grundlage für ... :		
<hr/>		
19. Medienform:		
<hr/>		
20. Angeboten von:	Forschungszentrum Simulationstechnologie	
<hr/>		

Modul: 10220 Modellierung

2. Modulkürzel:	052010001	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Dr. Frank Leymann		
9. Dozenten:	Bernhard Mitschang Frank Leymann		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.A. (L) Informatik HF, PO 079-1-2015, 5. Semester → Auswahl 2 --> Wahlmodule		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • Modul 10280 Programmierung und Software-Entwicklung • Modul 12060 Datenstrukturen und Algorithmen • Modul 40090 Systemkonzepte und -programmierung 		
12. Lernziele:	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.		
13. Inhalt:	<ul style="list-style-type: none"> • Entity-Relationship Modell und komplexe Objekte • Relationenmodell und Relationenalgebra , Überblick SQL - Transformationen von ER nach Relationen, Normalisierung • XML, DTD, XML-Schema, Info-Set, Namensräume • Metamodelle und Repository - RDF, RDF-S und 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 und Design, 2nd Edition, 1994. • H.J. Habermann, F. Leymann, Repository , Oldenbourg 1993. • W. Reising, Petri-Netze , Vieweg und Teubner 2010. • B. Silver, BPMN Method und Style ,Cody-Cassidy Press 2009. 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 102201 Vorlesung Modellierung • 102202 Übung Modellierung 		
16. Abschätzung Arbeitsaufwand:			
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 10221 Modellierung (PL), Schriftlich, 90 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich [10221] Modellierung (PL), schriftliche Prüfung, 90 Min., Gewicht: 1.0, Prüfungsvorleistung: Übungsschein [Prüfungsvorleistung] Vorleistung (USL-V), schriftlich, eventuell mündlich 		

18. Grundlage für ... : Architektur von Anwendungssystemen Datenbanken und Informationssysteme

19. Medienform:

20. Angeboten von: Architektur von Anwendungssystemen

Modul: 36100 Programmierparadigmen

2. Modulkürzel:	051510010	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch
8. Modulverantwortlicher:		Univ.-Prof. Dr. Erhard Plödereder	
9. Dozenten:		Erhard Plödereder	
10. Zuordnung zum Curriculum in diesem Studiengang:		B.A. (L) Informatik HF, PO 079-1-2015, 5. Semester → Auswahl 2 --> Wahlmodule	
11. Empfohlene Voraussetzungen:		Programmiererfahrung in mindestens einer Programmiersprache, vornehmlich Java, so wie z. B. im Modul "Programmierung und Software Entwicklung"(10280) erworben.	
12. Lernziele:		Die Studierenden haben grundlegende Konzepte von Programmiersprachen verstanden, die dem Erlernen weiterer Sprachen und dem vertieften Verständnis ihnen bekannter Sprachen dienlich sind. Sie haben deren Anwendung in mindestens einer weiteren Programmiersprache ihrer Ausbildung verstanden. Sie können ihre Kenntnisse in einfachen Programmen anwenden. Sie können weitere Programmiersprachen in ihrer akademischen und beruflichen Karriere schneller und präziser erlernen.	
13. Inhalt:		<p>Überblick typischer Konzepte in Programmiersprachen und ihrer Auswirkungen auf die Sprache und deren Anwendung:</p> <ul style="list-style-type: none"> • 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, Pearson Verlag, 2010 (Hörschein verfügbar). • 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:			
17. Prüfungsnummer/n und -name:		<ul style="list-style-type: none"> • 36101 Programmierparadigmen (PL), Schriftlich, 90 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich 	

[36101] Programmierparadigmen (PL), schriftliche Prüfung, 90 Min., Gewicht: 1.0 [Prüfungsvorleistung] Vorleistung (USL-V), schriftlich, eventuell mündlich

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Programmiersprachen und Übersetzerbau

Modul: 40090 Systemkonzepte und -programmierung

2. Modulkürzel:	051200005	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Dr. Kurt Rothermel		
9. Dozenten:	Kurt Rothermel Frank Dürr		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.A. (L) Informatik HF, PO 079-1-2015, 5. Semester → Auswahl 2 --> Wahlmodule		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • Modul 10280 Programmierung und Software-Entwicklung ; • Modul 12060 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 Modellierung und Analyse nebenläufiger Programme ; • Abstraktionen: Atomare Befehle, Prozesse, nebenläufiges Programm ; • Korrektheit- und Leitungskriterien Betriebssystemkonzepte ; • 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 Basisalgorithmen 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		

