

**Modulhandbuch**  
**Studiengang Künstlerisches Lehramt (GymPO I) Informatik**  
**Prüfungsordnung: 2010**  
Hauptfach

Wintersemester 2016/17  
Stand: 11. Oktober 2016

Universität Stuttgart  
Keplerstr. 7  
70174 Stuttgart

## Kontaktpersonen:

---

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

## Inhaltsverzeichnis

<b>Präambel</b> .....	<b>4</b>
<b>200 Pflichtmodule</b> .....	<b>5</b>
27620 Algorithmen und Berechenbarkeit für Lehramt .....	6
12060 Datenstrukturen und Algorithmen .....	7
14360 Einführung in die Technische Informatik .....	9
10190 Mathematik für Informatiker und Softwaretechniker .....	11
27630 Praktische Informatik für Lehramt .....	13
10280 Programmierung und Software-Entwicklung .....	14
10320 Seminar-INF 1 .....	16
27640 Systemkonzepte und -programmierung für Lehramt .....	18
10940 Theoretische Grundlagen der Informatik .....	19
<b>300 Wahlmodule</b> .....	<b>21</b>
10030 Architektur von Anwendungssystemen .....	22
10080 Datenbanken und Informationssysteme .....	24
10100 Grundlagen der Graphischen Ingenieursysteme .....	26
56930 Grundlagen der Rechnerarchitektur .....	27
<b>400 Fachdidaktikmodule</b> .....	<b>29</b>
34050 Gestaltung von Lehr- / Lernprozessen im Informatik-Unterricht, Projekt .....	30
34060 Grundlagen der Fachdidaktik Informatik .....	31
<b>3000 Zwischenprüfung</b> .....	<b>33</b>
12060 Datenstrukturen und Algorithmen .....	34
10280 Programmierung und Software-Entwicklung .....	36
10940 Theoretische Grundlagen der Informatik .....	38

## Präambel

Informatik ist die Wissenschaft von der Informationsverarbeitung und den informationsverarbeitenden Systemen. Sie umfasst deren Theorie und Methodik, den Einsatz dieser Systeme, aber auch die Auswirkungen. Die Informatik ist damit ein Grundpfeiler der modernen Informationsgesellschaft. Informatiksysteme durchdringen unser tägliches Leben. Was noch vor wenigen Jahren unvorstellbar war, ist heute selbstverständlicher Standard. Die weltweite freie Bereitstellung von Wissen und die Möglichkeit, sich ohne Kosten per E-Mail auszutauschen sowie riesige Datenmengen, etwa in Form von Musik und Filmen zu speichern, bedeutet eine gesellschaftliche Neuerung, an deren Gestaltung man durch ein Informatikstudium aktiv mitwirken kann.

Durch Verfahren der Modellbildung und Abstraktion formuliert die Informatik allgemeine Gesetze, die der Informationsverarbeitung zugrunde liegen, und sucht Standardlösungen für praxisrelevante Aufgaben. Von wachsender Bedeutung wird dabei die Beherrschung immer komplexer werdender verteilter und vernetzter Systeme. Informatikerinnen und Informatiker operieren mit abstrakten Zeichen und Objekten, untersuchen Daten-, Sprach- und Systemstrukturen und entwickeln formale Programmiersprachen zur Formulierung von Algorithmen, Prozessen, Systemen und speziellen Anwendungen. Die Hard- und Software-Systeme stehen dabei als Forschungsobjekte und gleichzeitig als Werkzeuge im Mittelpunkt der Arbeit. Durch Visualisierung und Simulation werden neue Anwendungen erschlossen. Informatik ist einerseits eine Strukturwissenschaft, andererseits dominieren aber heute die ingenieurwissenschaftlichen Methoden und Verfahren. Die Informatik an der Universität Stuttgart ist geprägt durch hohen Praxisbezug und Anwendungen, ohne dabei die notwendigen Grundlagen zu vernachlässigen.

## 200 Pflichtmodule

---

Zugeordnete Module:	10190	Mathematik für Informatiker und Softwaretechniker
	10280	Programmierung und Software-Entwicklung
	10320	Seminar-INF 1
	10940	Theoretische Grundlagen der Informatik
	12060	Datenstrukturen und Algorithmen
	14360	Einführung in die Technische Informatik
	27620	Algorithmen und Berechenbarkeit für Lehramt
	27630	Praktische Informatik für Lehramt
	27640	Systemkonzepte und -programmierung für Lehramt

---

## Modul: 27620 Algorithmen und Berechenbarkeit für Lehramt

2. Modulkürzel:	050420021	5. Moduldauer:	1 Semester
3. Leistungspunkte:	7.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Apl. Prof. Ulrich Hertrampf		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Stefan Funke</li> <li>• Volker Diekert</li> <li>• Ulrich Hertrampf</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:			
11. Empfohlene Voraussetzungen:	Vorlesungen aus dem 1. und 2. Semester		
12. Lernziele:	Die Studierenden kennen die Klassifizierung von Algorithmen in effizient berechenbar, NP-vollständig, PSPACE-Algorithmen und prinzipielle Unberechenbarkeit. Sie haben wichtige Entwurfsstrategien und Analysemethoden kennengelernt.		
13. Inhalt:	Berechenbarkeit vs. Unberechenbarkeit, Church'sche These, NP-Vollständigkeit, PSPACE-vollständige Algorithmen (QBF). Entwurfsstrategien: Teile und Herrsche, gierig (greedy), Dynamisches Programmieren, Randomisierte Algorithmen		
14. Literatur:	<ul style="list-style-type: none"> <li>• John Hopcroft, Jeffrey Ullman: Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie, 1988</li> <li>• Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein: Introduction to Algorithms (Second Edition), 2001</li> <li>• Volker Diekert: Entwurf und Analyse effizienter Algorithmen (Vorlesungsskript), 2006</li> </ul>		
15. Lehrveranstaltungen und -formen:	276201 Vorlesung mit Übungen Algorithmen und Berechenbarkeit		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- / Nacharbeitszeit:	168 h	
	Summe:	210 h	
17. Prüfungsnummer/n und -name:	27621 Algorithmen und Berechenbarkeit für Lehramt (LBP), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Formale Methoden der Informatik		

## Modul: 12060 Datenstrukturen und Algorithmen

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

	<ul style="list-style-type: none"> <li>• Algorithmenentwurf und -muster (inkrementell, greedy, divide-and-conquer, dynamische Programmierung, Backtracking, randomisierte Algorithmen)</li> <li>• Maschinelles Lernen (überwachtes Lernen, Entscheidungsbäume, SVM, neuronale Netze; unüberwachtes Lernen, k-Means)</li> </ul>								
14. Literatur:	<ul style="list-style-type: none"> <li>• G. Saake, K. Sattler. <i>Algorithmen und Datenstrukturen: Eine Einführung mit Java</i> . 5. Auflage, dpunkt-Verlag, 2013</li> <li>• T. Ottmann, P. Widmayer. <i>Algorithmen und Datenstrukturen</i> . 5. Auflage, Springer-Verlag, 2012</li> </ul>								
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 120601 Vorlesung Datenstrukturen und Algorithmen</li> <li>• 120602 Übung Datenstrukturen und Algorithmen</li> </ul>								
16. Abschätzung Arbeitsaufwand:	<table> <tr> <td>Präsenzzeit:</td> <td>63 h</td> </tr> <tr> <td>Selbststudiums- /</td> <td>207</td> </tr> <tr> <td>Nachbearbeitungszeit:</td> <td></td> </tr> <tr> <td>Summe:</td> <td>270 h</td> </tr> </table>	Präsenzzeit:	63 h	Selbststudiums- /	207	Nachbearbeitungszeit:		Summe:	270 h
Präsenzzeit:	63 h								
Selbststudiums- /	207								
Nachbearbeitungszeit:									
Summe:	270 h								
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 12061 Datenstrukturen und Algorithmen (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein. Die genauen Details der Übungsleistungen und Ihrer Anrechnung werden zu Beginn der Lehrveranstaltung bekannt gegeben.</li> <li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich</li> </ul>								
18. Grundlage für ... :									
19. Medienform:									
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme								

## Modul: 14360 Einführung in die Technische Informatik

2. Modulkürzel:	051400105	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Otto Eggenberger		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Otto Eggenberger</li> <li>• Sven Simon</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:			
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"> <li>• Informationsdarstellung</li> <li>• Zahlendarstellung und Codes</li> <li>• Digitale Grundbausteine</li> <li>• Logische Funktionen, Speicherelemente</li> <li>• Befehlsausführung, Programmablauf</li> </ul> <p>Elektrotechnische Grundlagen</p> <ul style="list-style-type: none"> <li>• Physikalische Grundbegriffe</li> <li>• Elektrische Spannung, elektrischer Strom</li> <li>• Elektrische Netzwerke</li> <li>• Halbleiterbauelemente</li> <li>• Digitale Grundschaltungen</li> </ul> <p>Digitale Schaltungen</p> <ul style="list-style-type: none"> <li>• Schaltnetzwerke</li> <li>• Boolesche Algebra und Schaltalgebra</li> <li>• Darstellung und Minimierung von Schaltfunktionen</li> <li>• Rückkopplung, Zustandsbegriff</li> <li>• Automaten und sequentielle Netzwerke</li> <li>• Digitale Standardschaltungen</li> <li>• Entwurfsmethodik</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Dirk W. Hoffman: Grundlagen der technischen Informatik, Hanser, 2007</li> <li>• Bernd Becker, Rolf Drechsler, Paul Molitor: Technische Informatik, Pearson Studium, 2005</li> <li>• Jörg Keller, Wolfgang J. Paul: Hardware Design, Teubner, 3. Aufl. 2005</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 143601 Vorlesung Einführung in die Technische Informatik</li> <li>• 143602 Gruppenübungen Einführung in die Technische Informatik</li> </ul>		

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

## Modul: 10190 Mathematik für Informatiker und Softwaretechniker

2. Modulkürzel:	080300100	5. Moduldauer:	2 Semester
3. Leistungspunkte:	18.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	12.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	PD Andreas Markus Kollross		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Peter Lesky</li> <li>• Wolfgang Rump</li> <li>• Wolf-Patrick Düll</li> <li>• Andreas Markus Kollross</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:			
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"> <li>• Grundlagen (Aussagenlogik, Mengen, Relationen, Abbildungen, Zahlenmengen, Grundbegriffe der Algebra)</li> <li>• Lineare Algebra (Vektorräume, lineare Abbildungen, Matrizen, Determinanten, lineare Gleichungssysteme, Eigenwerte, Normalformen, Hauptachsentransformation, Skalarprodukte)</li> <li>• Analysis (Konvergenz, Zahlenfolgen und Zahlenreihen, stetige Abbildungen, Folgen und Reihen von Funktionen, spezielle Funktionen)</li> </ul> <p>2. Semester:</p> <ul style="list-style-type: none"> <li>• Differential- und Integralrechnung (Funktionen einer und mehrerer Variablen, Ableitungen, Taylorentwicklungen, Extremwerte, Integration, Anwendungen)</li> <li>• Gewöhnliche Differentialgleichungen (elementar lösbare Differentialgleichungen, Existenz und Eindeutigkeit von Lösungen)</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Anna Sändig, Mathematik, Vorlesungskripte , SS 2007</li> <li>• D. Hachenberger, Mathematik für Informatiker, 2005</li> <li>• M. Brill, Mathematik für Informatiker, 2001</li> <li>• P.Hartmann, Mathematik für Informatiker, 2002</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 101901 Vorlesung Mathematik 1 für Informatik und Softwaretechnik</li> <li>• 101902 Übung Mathematik 1 für Informatik und Softwaretechnik</li> <li>• 101903 Vorlesung Mathematik 2 für Informatik und Softwaretechnik</li> <li>• 101904 Übung Mathematik 2 für Informatik und Softwaretechnik</li> </ul>		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit: 126 Stunden Nachbearbeitungszeit: 414 Stunden</p>		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 10191 Mathematik für Informatiker und Softwaretechniker (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Ein Übungsschein aus den beiden Veranstaltungen, jeweils im 1. oder 2. Fachsemester zu erwerben</li> </ul>		

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

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von:

---

## Modul: 27630 Praktische Informatik für Lehramt

2. Modulkürzel:	050420031	5. Moduldauer:	1 Semester
3. Leistungspunkte:	12.0 LP	6. Turnus:	jedes Semester
4. SWS:	8.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Stefan Funke		
9. Dozenten:	Dozenten der Informatik		
10. Zuordnung zum Curriculum in diesem Studiengang:			
11. Empfohlene Voraussetzungen:			
12. Lernziele:	Lernziele der zugehörigen Veranstaltungen		
13. Inhalt:	Inhalte der zugehörigen Veranstaltungen		
14. Literatur:	wird in den Veranstaltungen angegeben		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 276301 Vorlesung Praktische Informatik für Lehramt (Teil A)</li> <li>• 276302 Vorlesung mit Übungen Modellierung (Teil B1)</li> <li>• 276303 Vorlesung mit Übungen Mensch-Computer-Interaktion (Teil B2)</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Summe: 360 h		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 27631 Praktische Informatik für Lehramt B1 (PL), schriftlich, eventuell mündlich, Gewichtung: 1.0</li> <li>• 27632 Praktische Informatik für Lehramt B2 (PL), schriftlich, eventuell mündlich, Gewichtung: 1.0</li> <li>• 27633 Praktische Informatik für Lehramt, Projektschein (USL), schriftlich, eventuell mündlich, Gewichtung: 1.0</li> </ul>		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Formale Methoden der Informatik		

## Modul: 10280 Programmierung und Software-Entwicklung

2. Modulkürzel:	051520005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	9.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	6.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	Frank Leymann		
10. Zuordnung zum Curriculum in diesem Studiengang:			
11. Empfohlene Voraussetzungen:	Keine. Teilnahme an einem Vorkurs Java ist hilfreich aber nicht notwendig.		
12. Lernziele:	Die Teilnehmer haben einen Überblick über das Gebiet der Informatik. Sie haben die wichtigsten Konzepte einer höheren Programmiersprache und ihrer Verwendung verstanden und sind in der Lage, kleine Programme (bis zu einigen hundert Zeilen) zu analysieren und selbst zu konzipieren und zu implementieren. Sie kennen die Möglichkeiten, Daten- und Ablaufstrukturen zu entwerfen, zu beschreiben und zu codieren. Sie haben die Abstraktionskonzepte moderner Programmiersprachen verstanden. Sie kennen die Techniken und Notationen zur Definition kontextfreier Programmiersprachen und können damit arbeiten.		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Die Programmiersprache Java und die virtuelle Maschine</li> <li>• Objekte, Klassen, Schnittstellen, Blöcke, Programmstrukturen, Kontrakte</li> <li>• Klassenmodellierung mit der UML</li> <li>• Objekterzeugung und -ausführung</li> <li>• Boolesche Logik</li> <li>• Verzweigungen, Schleifen, Routinen, Abstraktionen, Modularisierung, Variablen, Zuweisungen</li> <li>• Rechner, Hardware</li> <li>• Syntaxdarstellungen</li> <li>• Übersicht über Programmiersprachen und -werkzeuge</li> <li>• Grundlegende Datenstrukturen und Algorithmen</li> <li>• Vererbung, Polymorphe</li> <li>• Semantik</li> <li>• Programmierung graphischer Oberflächen</li> <li>• Übergang zum Software Engineering</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Appelrath, Hans-Jürgen und Ludewig, Jochen, "Skriptum Informatik - eine konventionelle Einführung", Verlag der Fachvereine Zürich und B.G. Teubner Stuttgart, 4. Auflage 1999</li> <li>• Meyer, Bertrand, "Touch of Class", Springer-Verlag, 2009</li> <li>• Savitch, Walter, "Java. An Introduction to Problem Solving and Programming", Pearson, 6. Auflage, 2012</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 102801 Vorlesung Programmierung und Softwareentwicklung</li> <li>• 102802 Übung Programmierung und Softwareentwicklung</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	63 h	
	Selbststudiums- /	187 h	
	Nachbearbeitungszeit:		
	Prüfungsvorbereitung:	20 h	
	Summe:	270 h	

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

## Modul: 10320 Seminar-INF 1

2. Modulkürzel:	050420095	5. Moduldauer:	1 Semester
3. Leistungspunkte:	3.0 LP	6. Turnus:	jedes Semester
4. SWS:	2.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Stefan Wagner		
9. Dozenten:	Dozenten der Informatik		
10. Zuordnung zum Curriculum in diesem Studiengang:			
11. Empfohlene Voraussetzungen:	Basismodule der Informatik, darüber hinaus variabel: Je nach dem gewählten Seminarthema können Vorkenntnisse aus weiteren Vorlesungen benötigt werden.		
12. Lernziele:	Die Studierenden können sich mit wissenschaftlicher Originalliteratur auseinandersetzen, deren Kernaussagen rezipieren und sich ein spezielles Thema überwiegend im Selbststudium erarbeiten. Sie sind fähig relevante Daten zu sammeln und zu interpretieren und ihre Erkenntnisse einem Fach- und Laienpublikum verständlich zu präsentieren und auf Fragen aus dem Publikum angemessen und sachgerecht zu reagieren. Sie haben gelernt, sich mit einem wissenschaftlichen Thema über einen längeren Zeitraum hinweg auseinander zu setzen und eigenständig aktuelle Hintergrundinformation zu beschaffen. Sie haben generische Kompetenzen erworben, etwa aktiv an einer wissenschaftlichen Diskussion zu einem vorher bekannten Thema teilzunehmen und durch Fragen an den Vortragenden ihr Verständnis zu erweitern. Sie können eine Diskussion leiten und moderieren und sind befähigt, ihre Ergebnisse den Seminarteilnehmern vorzustellen und mit Hilfe moderner Präsentationstechniken zu visualisieren. Sie sind in der Lage, das von ihnen erarbeitete Thema auch schriftlich darzustellen.		
13. Inhalt:	<p>Variabel: Es werden Seminare zu diversen, häufig aktuellen Themen angeboten.</p> <p>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.</p>		
14. Literatur:	Die begleitende Literatur wird in der Veranstaltung und im Web bekannt gegeben.		
15. Lehrveranstaltungen und -formen:	103201 Seminar		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	21 h	
	Selbststudiums- /	69 h	
	Nachbearbeitungszeit:		
	Summe:	90 h	
17. Prüfungsnummer/n und -name:	10321 Seminar-INF 1 (LBP), Sonstiges, Gewichtung: 1.0, Scheinkriterien sind in der Regel ein Vortrag, eine schriftliche		

Ausarbeitung, sowie die aktive Mitarbeit während der Seminarveranstaltung.

---

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von: Institut für Softwaretechnologie

---

## Modul: 27640 Systemkonzepte und -programmierung für Lehramt

---

2. Modulkürzel:	051200095	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch

---

8. Modulverantwortlicher:	Univ.-Prof. Kurt Rothermel
9. Dozenten:	<ul style="list-style-type: none"><li>• Kurt Rothermel</li><li>• Frank Dürr</li></ul>

---

10. Zuordnung zum Curriculum in diesem Studiengang:

---

11. Empfohlene Voraussetzungen:

---

12. Lernziele:

---

13. Inhalt:

---

14. Literatur:

---

15. Lehrveranstaltungen und -formen:	276401 Vorlesung mit Übungen Systemkonzepte und -programmierung
--------------------------------------	---

---

16. Abschätzung Arbeitsaufwand:

---

17. Prüfungsnummer/n und -name:	27641 Systemkonzepte und -programmierung für Lehramt (LBP), schriftliche Prüfung, 120 Min., Gewichtung: 1.0
---------------------------------	---

---

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von:

---

## Modul: 10940 Theoretische Grundlagen der Informatik

2. Modulkürzel:	050420005	5. Moduldauer:	2 Semester
3. Leistungspunkte:	12.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	8.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Apl. Prof. Ulrich Hertrampf		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Volker Diekert</li> <li>• Ulrich Hertrampf</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:			
11. Empfohlene Voraussetzungen:	Keine		
12. Lernziele:	<p>Logik und Diskrete Strukturen:</p> <ul style="list-style-type: none"> <li>• Die Studierenden haben die grundsätzlichen Kenntnisse in Logik und Diskreter Mathematik erworben, wie sie in den weiteren Grundvorlesungen der Informatik in verschiedenen Bereichen benötigt werden.</li> </ul> <p>Automaten und Formale Sprachen:</p> <ul style="list-style-type: none"> <li>• Die Studierenden beherrschen wichtige theoretische Grundlagen der Informatik, insbesondere die Theorie und Algorithmik endlicher Automaten. Hierzu gehört das Kennenlernen, Einordnung und Trennung der Chomskyschen Sprachklassen.</li> </ul>		
13. Inhalt:	<p>Logik und Diskrete Strukturen:</p> <ul style="list-style-type: none"> <li>• Einführung in die Aussagenlogik: Semantik (Wahrheitswerte), Syntax (Axiome und Schlussregeln), Normalformen; Hornformeln; Endlichkeitssatz; aussagenlogische Resolution;</li> <li>• Einführung in die Prädikatenlogik 1. Stufe: Semantik und Syntax, Normalformen, Unifikatoren, Herbrand-Theorie, prädikatenlogische Resolution;</li> <li>• Elementare Zahlentheorie: Rechnen mit Restklassen, endliche Körper, Euklidischer Algorithmus, Chinesischer Restsatz, Primzahltests, RSA-Verfahren; Wachstumsabschätzungen; Grundbegriffe der Wahrscheinlichkeitsrechnung; Kombinatorik; Graphen.</li> </ul> <p>Automaten und Formale Sprachen:</p> <ul style="list-style-type: none"> <li>• Deterministische- bzw. nichtdeterministische endliche Automaten, reguläre Ausdrücke, Minimierung endlicher Automaten, Iterationslemmata für reguläre und kontextfreie Sprachen, Normalformen, Kellerautomaten, Lösen des Wortproblems kontextfreier Sprachen mit dem CYK-Algorithmus, linear beschränkte Automaten, kontextsensitive Grammatiken, Typ 0-Grammatiken und Turingmaschinen.</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• John Hopcroft, Jeffrey Ullman, Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie, 1988</li> <li>• Uwe Schöning, Theoretische Informatik - kurzgefasst, 1999</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 109401 Vorlesung Logik und Diskrete Strukturen</li> <li>• 109402 Übung Logik und Diskrete Strukturen</li> </ul>		

- 109403 Vorlesung Automaten und Formale Sprachen
  - 109404 Übung Automaten und Formale Sprachen
- 

16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	84 h
	Selbststudiums- /	276 h
	Nachbearbeitungszeit:	
	Summe:	360 h

---

17. Prüfungsnummer/n und -name:	• 10941 Theoretische Grundlagen der Informatik (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein
	• V Vorleistung (USL-V), schriftlich, eventuell mündlich, 30 Min.

---

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von: Institut für Formale Methoden der Informatik

---

## 300 Wahlmodule

---

Zugeordnete Module:    10030 Architektur von Anwendungssystemen  
                              10080 Datenbanken und Informationssysteme  
                              10100 Grundlagen der Graphischen Ingenieursysteme  
                              56930 Grundlagen der Rechnerarchitektur

---

## Modul: 10030 Architektur von Anwendungssystemen

2. Modulkürzel:	052010002	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	Frank Leymann		
10. Zuordnung zum Curriculum in diesem Studiengang:			
11. Empfohlene Voraussetzungen:	Vorlesungen des Grundstudiums.		
12. Lernziele:	Die Vorlesung erläutert den Begriff der Architektur von Anwendungssystemen und die Rolle des Architekten solcher Systeme. Die wesentlichen Bestandteile von Anwendungsarchitektur wie etwa Datenbanksysteme, Anwendungsserver, Messaging Systeme, Workflowsysteme und TP-Monitore werden diskutiert. Die wesentlichen Mustern zur Erstellung von Anwendungssystemen sind verstanden.		
13. Inhalt:	Architekturelle Stile wie etwa N-stufige Aufbauten oder Service-Orientierung werden vorgestellt. Architekturmuster werden detailliert. Fundamentale Konzepte wie Transaktionen und Queuing werden eingeführt. Darauf aufbauend wird Direct TP vs Queues TP diskutiert. Grundlegende Qualitätseigenschaften wie Verfügbarkeit und Skalierbarkeit werden erläutert und Mechanismen zu deren Erzielen eingeführt. Die Rolle von Komponenten und Programmierung im Großen wird heraus gearbeitet und Modell-getriebene Architektur vorgestellt.		
14. Literatur:	<ul style="list-style-type: none"> <li>• A. Silberschatz, H. F. Korth, S. Sudarshan, Database System Concepts, 2002</li> <li>• B. Neubauer, T. Ritter, F. Stoinnski, CORBA Komponenten, 2004</li> <li>• F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal, Pattern-orientierte Software Architektur - Ein Patternsystem, 1998</li> <li>• F. Leymann, D. Roller, Production Workflow, 2000</li> <li>• L. Hohmann, Beyond Software Architecture, 2003</li> <li>• M. Fowler, Patterns of Enterprise Application Architecture, 2003</li> <li>• P. Bernstein, E. Newcomer, Principles of Transaction Processing, 1997</li> <li>• S. Conrad, W. Hasselbring, A. Koschel, R. Tritsch, Enterprise Application Integration, 2006</li> <li>• S. Weerawarana, F. Curbera, F. Leymann, T. Storey, D. Ferguson, Web Services Platform Architecture, 2005</li> <li>• W. Emmerich, Konstruktion von verteilten Objekten, 2003</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 100301 Vorlesung Grundlagen der Architektur von Anwendungssystemen</li> <li>• 100302 Übung Grundlagen der Architektur von Anwendungssystemen</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		

17. Prüfungsnummer/n und -name: 10031 Architektur von Anwendungssystemen (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0,

---

18. Grundlage für ... :  
• 29530 Business Process Management  
• 29480 Loose Coupling and Message Based Applications  
• 29510 Service Computing

---

19. Medienform: Vorlesungen mit begleitenden Übungen

---

20. Angeboten von: Architektur von Anwendungssystemen

---

## Modul: 10080 Datenbanken und Informationssysteme

2. Modulkürzel:	051200025	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Bernhard Mitschang		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Bernhard Mitschang</li> <li>• Holger Schwarz</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:			
11. Empfohlene Voraussetzungen:	Vorlesung Modellierung oder Gleichwertiges		
12. Lernziele:	Die Studierenden haben die erforderlichen Kenntnisse für Datenbankprogrammierer in angemessenem Umfang erworben.		
13. Inhalt:	<p>Die Vorlesung "Datenbanken und Informationssysteme" ist als Einstiegsveranstaltung in das Vertiefungsgebiet Datenbanksysteme konzipiert. Aufbauend auf dem Inhalt der Vorlesung "Modellierung" werden insbesondere Entwurfs- und Realisierungsaspekte von Datenbanksystemen betrachtet. Die Entwicklung, Installation und Administration von Datenbanksystemen bestimmen hier sowohl Stoffauswahl als auch Detaillierungsgrad.</p> <p>Als Grundlage für alle weiteren Betrachtungen wird ein Schichtenmodell zur Beschreibung eines allgemeinen Datenbanksystems vorgestellt. Darauf aufbauend werden die einzelnen Systemschichten im Detail diskutiert, die dort zu realisierenden Komponenten betrachtet sowie die jeweils vorherrschenden Algorithmen beschrieben und bewertet. Im Einzelnen werden folgende Aspekte vertieft: Anwendungsprogrammierschnittstelle, Externspeicherverwaltung, DBS-Pufferverwaltung, Speicherungsstrukturen und Zugriffspfadstrukturen, Anfrageverarbeitung und Anfrageoptimierung, Transaktionsverarbeitung, Synchronisation, Logging und Recovery.</p>		
14. Literatur:	<ul style="list-style-type: none"> <li>• A. Kemper, A. Eickler, Datenbanksysteme - Eine Einführung, 2004</li> <li>• Th. Härder, E. Rahm, Datenbanksysteme, 2008</li> <li>• H. Garcia-Molina, J. D. Ullman, J. Widom, Database Systems. The Complete Book, 2003</li> <li>• R. Elmasri, S. Navathe, Fundamentals of Database Systems, 2003</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 100801 Vorlesung Datenbanken und Informationssysteme</li> <li>• 100802 Übung Datenbanken und Informationssysteme</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 Stunden	
	Selbststudium:	138 Stunden	
17. Prüfungsnummer/n und -name:	10081 Datenbanken und Informationssysteme (PL), schriftlich oder mündlich, 60 Min., Gewichtung: 1.0, Prüfungsvorleistung: Modalitäten werden in der ersten Vorlesung angegeben		
18. Grundlage für ... :			

19. Medienform:

---

20. Angeboten von:

---

## Modul: 10100 Grundlagen der Graphischen Ingenieursysteme

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

## Modul: 56930 Grundlagen der Rechnerarchitektur

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Hans-Joachim Wunderlich		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Michael Kochte</li> <li>• Hans-Joachim Wunderlich</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:			
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 41930 Rechnerorganisation</li> </ul>		
12. Lernziele:	<ul style="list-style-type: none"> <li>• Tiefes Verständnis von Entwurfskonzepten, die in modernen Prozessoren und Rechensystemen Verwendung finden</li> <li>• Kenntnis von Entwurfsherausforderungen.</li> <li>• Verständnis von aktuellen und zukünftigen Entwicklungstrends</li> </ul>		
13. Inhalt:	<p>Grundlegende und fortgeschrittene Themen der Rechnerarchitektur, inklusive:</p> <ul style="list-style-type: none"> <li>• Technologiegrundlagen: Entwurfsverfahren, Herstellungsmethoden, Ausbeute, Test und Zuverlässigkeit, Kosten und Qualität, Skalierung.</li> <li>• Rechen- und Verlustleistung: Analyse und Optimierung</li> <li>• Rechnerarithmetik: Effiziente Hardwarestrukturen für grundlegende Arithmetik, Implementierung von Logarithmen, Exponentialfunktion und trigonometrischen Funktionen, arithmetische Pipelines, praktische Implementierungen von Gleitkommaarithmetik (Cell SPE, SPARC).</li> <li>• Instruktionsparallelismus (ILP): Superskalarität, statisches und dynamisches Scheduling, out-of-order execution, VLIW Prozessoren, Multithreading.</li> <li>• Datenparallelismus (DLP): Vektorprozessoren, SIMD, Grafikprozessoreinheiten (GPGPU)</li> <li>• Threadparallelismus (TLP): Mehrprozessorsysteme, Speicherkohärenz und Synchronisierung.</li> <li>• Speicher- und Cache-Architekturen: Entwurf und Optimierung</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• J. L. Hennessy, D. A. Patterson, Computer Architecture: A Quantitative Approach, 2012</li> <li>• I. Koren, Computer Arithmetic Algorithms, 2001</li> <li>• Powerpoint Foliensatz</li> <li>• Auswahl von wissenschaftlichen Artikeln</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 569301 Vorlesung Grundlagen der Rechnerarchitektur</li> <li>• 569302 Übung Grundlagen der Rechnerarchitektur</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	56931 Grundlagen der Rechnerarchitektur (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			

20. Angeboten von: Rechnerarchitektur

---

## 400 Fachdidaktikmodule

---

Zugeordnete Module:   34050 Gestaltung von Lehr- / Lernprozessen im Informatik-Unterricht, Projekt  
                              34060 Grundlagen der Fachdidaktik Informatik

---

## Modul: 34050 Gestaltung von Lehr- / Lernprozessen im Informatik-Unterricht, Projekt

---

2. Modulkürzel:	101010072	5. Moduldauer:	1 Semester
3. Leistungspunkte:	4.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	2.0	7. Sprache:	Deutsch

---

8. Modulverantwortlicher:	Apl. Prof. Ulrich Hertrampf
9. Dozenten:	Bernd Zinn

---

10. Zuordnung zum Curriculum in diesem Studiengang:

---

11. Empfohlene Voraussetzungen:

---

12. Lernziele:

---

13. Inhalt:

---

14. Literatur:

---

15. Lehrveranstaltungen und -formen:	340501 Seminar Gestaltung von Lehr- / Lernprozessen im Informatik-Unterricht, Projekt
--------------------------------------	---

---

16. Abschätzung Arbeitsaufwand:

---

17. Prüfungsnummer/n und -name:	34051 Gestaltung von Lehr- / Lernprozessen im Informatik-Unterricht, Projekt (PL), schriftlich, eventuell mündlich, Gewichtung: 1.0
---------------------------------	---

---

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von:	Institut für Erziehungswissenschaft
--------------------	-------------------------------------

---

## Modul: 34060 Grundlagen der Fachdidaktik Informatik

2. Modulkürzel:	101010062	5. Moduldauer:	2 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Bernd Zinn		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Andreas Mußotter</li> <li>• Bernd Zinn</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:			
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"> <li>• <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></li> <li>• <i>methodisch-didaktische Ansätze im technischen Unterricht, Berufsfeldspezifische Aspekte (z.B. Lernen in technischen Reallernräumen, Experimente)</i></li> <li>• <i>Umgang mit Inklusion und Heterogenität,</i></li> <li>• <i>Pädagogische Diagnostik</i></li> <li>• <i>Analyse berufs- und schulformbezogener Lehrpläne</i></li> <li>• <i>Planung, Durchführung und Evaluation von technischem Unterricht in der Aus- und Weiterbildung</i></li> </ul>		

- *Wandel beruflicher Anforderungen und Rahmenbedingungen in der Informatik*
- *Aktuelle Inhalte der Lehr-Lernforschung im Bezugfeld der Technikdidaktik und speziell Fachdidaktik Informatik*

---

14. Literatur:

- *Schubert, S. & 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. & 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 Bezugfeld 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, eventuell mündlich, Gewichtung: 1.0
- 34062 Grundlagen der Fachdidaktik Informatik, Ausarbeitung incl. Präsentation (USL), schriftlich, eventuell mündlich, Gewichtung: 1.0

---

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von:

---

## 3000 Zwischenprüfung

---

Zugeordnete Module:   10280 Programmierung und Software-Entwicklung  
                          10940 Theoretische Grundlagen der Informatik  
                          12060 Datenstrukturen und Algorithmen

---

## Modul: 12060 Datenstrukturen und Algorithmen

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

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

---

14. Literatur:	<ul style="list-style-type: none"> <li>• G. Saake, K. Sattler. <i>Algorithmen und Datenstrukturen: Eine Einführung mit Java</i> . 5. Auflage, dpunkt-Verlag, 2013</li> <li>• T. Ottmann, P. Widmayer. <i>Algorithmen und Datenstrukturen</i> . 5. Auflage, Springer-Verlag, 2012</li> </ul>								
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 120601 Vorlesung Datenstrukturen und Algorithmen</li> <li>• 120602 Übung Datenstrukturen und Algorithmen</li> </ul>								
16. Abschätzung Arbeitsaufwand:	<table border="0" style="width: 100%;"> <tr> <td style="width: 60%;">Präsenzzeit:</td> <td style="text-align: right;">63 h</td> </tr> <tr> <td>Selbststudiums- /</td> <td style="text-align: right;">207</td> </tr> <tr> <td>Nachbearbeitungszeit:</td> <td></td> </tr> <tr> <td>Summe:</td> <td style="text-align: right;">270 h</td> </tr> </table>	Präsenzzeit:	63 h	Selbststudiums- /	207	Nachbearbeitungszeit:		Summe:	270 h
Präsenzzeit:	63 h								
Selbststudiums- /	207								
Nachbearbeitungszeit:									
Summe:	270 h								
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 12061 Datenstrukturen und Algorithmen (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein. Die genauen Details der Übungsleistungen und Ihrer Anrechnung werden zu Beginn der Lehrveranstaltung bekannt gegeben.</li> <li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich</li> </ul>								
18. Grundlage für ... :									
19. Medienform:									
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme								

---

## Modul: 10280 Programmierung und Software-Entwicklung

2. Modulkürzel:	051520005	5. Moduldauer:	1 Semester										
3. Leistungspunkte:	9.0 LP	6. Turnus:	jedes 2. Semester, WiSe										
4. SWS:	6.0	7. Sprache:	Deutsch										
8. Modulverantwortlicher:		Univ.-Prof. Frank Leymann											
9. Dozenten:		Frank Leymann											
10. Zuordnung zum Curriculum in diesem Studiengang:													
11. Empfohlene Voraussetzungen:		Keine. Teilnahme an einem Vorkurs Java ist hilfreich aber nicht notwendig.											
12. Lernziele:		Die Teilnehmer haben einen Überblick über das Gebiet der Informatik. Sie haben die wichtigsten Konzepte einer höheren Programmiersprache und ihrer Verwendung verstanden und sind in der Lage, kleine Programme (bis zu einigen hundert Zeilen) zu analysieren und selbst zu konzipieren und zu implementieren. Sie kennen die Möglichkeiten, Daten- und Ablaufstrukturen zu entwerfen, zu beschreiben und zu codieren. Sie haben die Abstraktionskonzepte moderner Programmiersprachen verstanden. Sie kennen die Techniken und Notationen zur Definition kontextfreier Programmiersprachen und können damit arbeiten.											
13. Inhalt:		<ul style="list-style-type: none"> <li>• Die Programmiersprache Java und die virtuelle Maschine</li> <li>• Objekte, Klassen, Schnittstellen, Blöcke, Programmstrukturen, Kontrakte</li> <li>• Klassenmodellierung mit der UML</li> <li>• Objekterzeugung und -ausführung</li> <li>• Boolesche Logik</li> <li>• Verzweigungen, Schleifen, Routinen, Abstraktionen, Modularisierung, Variablen, Zuweisungen</li> <li>• Rechner, Hardware</li> <li>• Syntaxdarstellungen</li> <li>• Übersicht über Programmiersprachen und -werkzeuge</li> <li>• Grundlegende Datenstrukturen und Algorithmen</li> <li>• Vererbung, Polymorphe</li> <li>• Semantik</li> <li>• Programmierung graphischer Oberflächen</li> <li>• Übergang zum Software Engineering</li> </ul>											
14. Literatur:		<ul style="list-style-type: none"> <li>• Appelrath, Hans-Jürgen und Ludewig, Jochen, "Skriptum Informatik - eine konventionelle Einführung", Verlag der Fachvereine Zürich und B.G. Teubner Stuttgart, 4. Auflage 1999</li> <li>• Meyer, Bertrand, "Touch of Class", Springer-Verlag, 2009</li> <li>• Savitch, Walter, "Java. An Introduction to Problem Solving and Programming", Pearson, 6. Auflage, 2012</li> </ul>											
15. Lehrveranstaltungen und -formen:		<ul style="list-style-type: none"> <li>• 102801 Vorlesung Programmierung und Softwareentwicklung</li> <li>• 102802 Übung Programmierung und Softwareentwicklung</li> </ul>											
16. Abschätzung Arbeitsaufwand:		<table> <tr> <td>Präsenzzeit:</td> <td>63 h</td> </tr> <tr> <td>Selbststudiums- /</td> <td>187 h</td> </tr> <tr> <td>Nachbearbeitungszeit:</td> <td></td> </tr> <tr> <td>Prüfungsvorbereitung:</td> <td>20 h</td> </tr> <tr> <td>Summe:</td> <td>270 h</td> </tr> </table>		Präsenzzeit:	63 h	Selbststudiums- /	187 h	Nachbearbeitungszeit:		Prüfungsvorbereitung:	20 h	Summe:	270 h
Präsenzzeit:	63 h												
Selbststudiums- /	187 h												
Nachbearbeitungszeit:													
Prüfungsvorbereitung:	20 h												
Summe:	270 h												

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

## Modul: 10940 Theoretische Grundlagen der Informatik

2. Modulkürzel:	050420005	5. Moduldauer:	2 Semester
3. Leistungspunkte:	12.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	8.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Apl. Prof. Ulrich Hertrampf		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Volker Diekert</li> <li>• Ulrich Hertrampf</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:			
11. Empfohlene Voraussetzungen:	Keine		
12. Lernziele:	<p>Logik und Diskrete Strukturen:</p> <ul style="list-style-type: none"> <li>• Die Studierenden haben die grundsätzlichen Kenntnisse in Logik und Diskreter Mathematik erworben, wie sie in den weiteren Grundvorlesungen der Informatik in verschiedenen Bereichen benötigt werden.</li> </ul> <p>Automaten und Formale Sprachen:</p> <ul style="list-style-type: none"> <li>• Die Studierenden beherrschen wichtige theoretische Grundlagen der Informatik, insbesondere die Theorie und Algorithmik endlicher Automaten. Hierzu gehört das Kennenlernen, Einordnung und Trennung der Chomskyschen Sprachklassen.</li> </ul>		
13. Inhalt:	<p>Logik und Diskrete Strukturen:</p> <ul style="list-style-type: none"> <li>• Einführung in die Aussagenlogik: Semantik (Wahrheitswerte), Syntax (Axiome und Schlussregeln), Normalformen; Hornformeln; Endlichkeitssatz; aussagenlogische Resolution;</li> <li>• Einführung in die Prädikatenlogik 1. Stufe: Semantik und Syntax, Normalformen, Unifikatoren, Herbrand-Theorie, prädikatenlogische Resolution;</li> <li>• Elementare Zahlentheorie: Rechnen mit Restklassen, endliche Körper, Euklidischer Algorithmus, Chinesischer Restsatz, Primzahltests, RSA-Verfahren; Wachstumsabschätzungen; Grundbegriffe der Wahrscheinlichkeitsrechnung; Kombinatorik; Graphen.</li> </ul> <p>Automaten und Formale Sprachen:</p> <ul style="list-style-type: none"> <li>• Deterministische- bzw. nichtdeterministische endliche Automaten, reguläre Ausdrücke, Minimierung endlicher Automaten, Iterationslemmata für reguläre und kontextfreie Sprachen, Normalformen, Kellerautomaten, Lösen des Wortproblems kontextfreier Sprachen mit dem CYK-Algorithmus, linear beschränkte Automaten, kontextsensitive Grammatiken, Typ 0-Grammatiken und Turingmaschinen.</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• John Hopcroft, Jeffrey Ullman, Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie, 1988</li> <li>• Uwe Schöning, Theoretische Informatik - kurzgefasst, 1999</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 109401 Vorlesung Logik und Diskrete Strukturen</li> <li>• 109402 Übung Logik und Diskrete Strukturen</li> </ul>		

- 109403 Vorlesung Automaten und Formale Sprachen
  - 109404 Übung Automaten und Formale Sprachen
- 

16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	84 h
	Selbststudiums- /	276 h
	Nachbearbeitungszeit:	
	Summe:	360 h

---

- |                                 |   |
|---------------------------------|---|
| 17. Prüfungsnummer/n und -name: | • 10941 Theoretische Grundlagen der Informatik (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein |
|                                 | • V Vorleistung (USL-V), schriftlich, eventuell mündlich, 30 Min.   |
- 

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von: Institut für Formale Methoden der Informatik

---