

---

**Modulhandbuch**  
**Informatik - Zwei-Fächer-Bachelor-Studiengang**  
**im Sommersemester 2024**  
erstellt am 17.07.2024

---

<b>inf030 - Programmierung, Datenstrukturen und Algorithmen</b>	4
<b>inf031 - Objektorientierte Modellierung und Programmierung</b>	7
<b>inf200 - Grundlagen der Technischen Informatik</b>	10
<b>inf700 - Didaktik der Informatik I</b>	12
<b>mat950 - Mathematik für Informatik (Diskrete Strukturen)</b>	14
<b>inf005 - Softwaretechnik I</b>	15
<b>inf007 - Informationssysteme I</b>	19
<b>inf700 - Didaktik der Informatik I</b>	21
<b>inf004 - Softwareprojekt</b>	23
<b>inf009 - Praktikum Datenbanken</b>	26
<b>inf014 - Praktikum Betriebssysteme</b>	28
<b>inf018 - Medienverarbeitung</b>	30
<b>inf021 - Fortgeschrittene Java-Technologien</b>	33
<b>inf202 - Praktikum Technische Informatik</b>	35
<b>inf800 - Proseminar Informatik</b>	38
<b>inf803 - Spezielle Themen der Informatik I</b>	40
<b>inf804 - Spezielle Themen der Informatik II</b>	42
<b>inf808 - Aktuelle Themen der Informatik</b>	44
<b>inf200 - Grundlagen der Technischen Informatik</b>	46
<b>inf201 - Technische Informatik</b>	48
<b>inf203 - Embedded Systems I</b>	50

---

<b>inf204 - Embedded Systems II</b>	53
<b>inf205 - Formale Methoden Eingebetteter Systeme</b>	55
<b>inf207 - Grundlagen der Elektrotechnik</b>	58
<b>inf208 - Mikrorobotik und Mikrosystemtechnik</b>	60
<b>inf209 - Regelungstechnik</b>	62
<b>inf210 - Signal- und Bildverarbeitung</b>	64
<b>inf400 - Theoretische Informatik: Logik</b>	66
<b>inf401 - Grundlagen der Theoretischen Informatik</b>	68
<b>inf407 - Programmverifikation</b>	70
<b>bam - Bachelorarbeitsmodul</b>	72

## Basismodule

### inf030 - Programmierung, Datenstrukturen und Algorithmen

<b>Modulbezeichnung</b>	Programmierung, Datenstrukturen und Algorithmen
<b>Modulkürzel</b>	inf030
<b>Kreditpunkte</b>	9,0 KP
<b>Workload</b>	270 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"> <li>• Fach-Bachelor Informatik (Bachelor) &gt; Basismodule</li> <li>• Fach-Bachelor Mathematik (Bachelor) &gt; Nebenfachmodule</li> <li>• Fach-Bachelor Nachhaltigkeitsökonomik (Bachelor) &gt; Wahlpflichtbereich</li> <li>• Fach-Bachelor Wirtschaftsinformatik (Bachelor) &gt; Basiscurriculum</li> <li>• Fach-Bachelor Wirtschaftswissenschaften (Bachelor) &gt; Studienrichtung Wirtschaftsinformatik</li> <li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Basismodule</li> </ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"> <li>• Schönberg, Christian (Modulverantwortung)</li> <li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li> </ul>
<b>Teilnahmevoraussetzungen</b>	Keine fachspezifischen Vorkenntnisse erforderlich.

#### Kompetenzziele

Das Programmieren ist eine der Basistätigkeiten von Informatikern (m/w/d) und Voraussetzung für viele andere Veranstaltungen des Informatikstudiums. Ziel des Moduls „Programmierung, Datenstrukturen und Algorithmen“ ist das Erlernen grundlegender Konzepte der imperativen, prozeduralen und objektorientierten Programmierung anhand der Programmiersprache Java sowie die Vorstellung bekannter, effizienter Algorithmen und Datenstrukturen für verschiedene, häufig vorkommende Problemstellungen.

Die Studierenden sollen nach Abschluss des Moduls selbstständig imperative und einfache objektorientierte Programme auf der Grundlage von Java für die Lösung kleinerer Probleme entwickeln und die Effizienz ihrer Programme einschätzen können. Außerdem sollen sie wichtige Algorithmen anwenden und aufgrund ihrer Komplexität auswählen können.

#### Fachkompetenzen

Die Studierenden

- beschreiben grundlegende Konzepte der imperativen Programmierung mit Java
- erkennen die Terminologie der imperativen Programmierung und verwenden die entsprechenden Begriffe präzise bei Diskussionen
- erkennen grundlegende Terminologie der objektorientierten Programmierung
- beschreiben, was ihnen vorgelegte Programme tun
- entwickeln selbstständig Programme für die Lösung kleinerer Probleme
- untersuchen systematisch eigene und fremde Programme auf Fehler
- setzen moderne Programmierungsumgebungen zum Entwickeln und Testen von Programmen ein
- erstellen Algorithmen mit allgemeinen Entwurfskonzepten (z.B. Greedy-Verfahren, Divide-and-Conquer-Verfahren)
- benennen Algorithmen und Datenstrukturen zur Lösung von häufig vorkommenden Problemen und bewerten diese in ihrer Anwendbarkeit
- benennen Probleme der Effizienz von algorithmischen Lösungen konkreter Fragestellungen und bewerten diese
- wählen fundiert einen Algorithmus und eine Datenstruktur zur Lösung eines konkreten Problems aus
- wenden die gelernten Algorithmen und Datenstrukturen sinnvoll auf gegebene und konkrete Probleme an

#### Methodenkompetenzen

---

Die Studierenden

- lösen gegebene Probleme unter den Gesichtspunkt der imperativen bzw. objektorientierten Programmierung
- übertragen praktische Erfahrungen in der Programmentwicklung auf neue Aufgaben

### **Sozialkompetenzen**

Die Studierenden

- vermitteln die Struktur und Wirkungsweise selbst entwickelter Programme an andere
- präsentieren selbstständig entwickelte Lösungen zu kleinen Aufgaben vor Gruppen

### **Selbstkompetenzen**

Die Studierenden

- organisieren sich beim Finden von algorithmischen Lösungen für kleine und mittelgroße Probleme der Informatik
- beziehen die Konzepte des allgemeinen Programmentwurfs in ihr Handeln ein

---

## **Modulinhalte**

Im ersten Teil werden allgemeine Grundbegriffe der Programmierung eingeführt:

- Algorithmus, Programmiersprachen, Computer
- Entwicklungswerkzeuge, Entwicklungsphasen
- Compiler
- Grammatiken
- Logik

Der zweite Teil befasst sich mit grundlegenden Programmierkonzepten:

- Datentypen
- Variablen
- Ausdrücke, Anweisungen
- Kontrollstrukturen
- Methoden, Parameter
- Rekursion
- Referenzdatentypen, Arrays
- Klassen, Objekte
- Dokumentation
- Testen

Der dritte Teil beinhaltet eine Einführung in Datenstrukturen und Algorithmen sowie die Diskussion ihrer Effizienz, d.h. des Berechnungsaufwands bzw. des Speicherbedarfs in Abhängigkeit vom Umfang der zu verarbeitenden Daten. Das Modul stellt für verschiedene, häufig vorkommende Problemstellungen bekannte, effiziente Algorithmen und Datenstrukturen vor. Dazu gehören insbesondere:

- Verfahren zum Suchen nach Schlüsseln, sowie Einfügen und Löschen in dynamischen Datenmengen, z.B. Listen, Bäume, AVL-Bäume oder Hash-Verfahren,
- Methoden zur Suche nach Textmustern,
- Verfahren zum Sortieren von Daten nach Schlüsselwerten, z.B. QuickSort und HeapSort,
- Graph-basierte Anwendungen, z.B. zur Ermittlung kürzester Wege in Graphen.

Ergänzt wird der Vorlesungsteil um einen umfassenden Übungsteil, in dem insbesondere die vermittelten Programmierinhalte an praktischen Beispielen umgesetzt werden.

## Literaturempfehlungen

### Essenziell

- Skript (wird in elektronischer Form über das Stud.IP fortlaufend in der Vorlesung zur Verfügung gestellt)

### Empfohlene Sekundärliteratur

- Dietmar Ratz, Jens Scheffler, Detlev Seese, Jan Wiesenberger: Grundkurs Programmieren in Java, Carl Hanser Verlag.
- Joachim Goll, Cornelia Heinish: Java als erste Programmiersprache, Springer Vieweg Verlag
- Ottmann, Widmayer: Algorithmen und Datenstrukturen. Spektrum Verlag, 5. Auflage, 2012
- Sedgewick, Wayne: Algorithms. Addison Wesley, 4th ed., 2011
- Siege: Einführung in die Informatik. Shaker Verlag, 2013

---

### Links

<b>Unterrichtssprache</b>	Deutsch	
<b>Dauer in Semestern</b>	1 Semester	
<b>Angebotsrhythmus Modul</b>	jedes Wintersemester	
<b>Aufnahmekapazität Modul</b>	unbegrenzt	
<b>Lehr-/Lernform</b>	1VL + 1Ü	
Prüfung	Prüfungszeiten	Prüfungsform
<b>Gesamtmodul</b>	Am Ende der Veranstaltungszeit	Klausur / Portfolio (schriftliche Kurzprüfungen) / mündliche Prüfung

---

Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenz
Vorlesung		4	WiSe	56
Übung		2	WiSe	28
<b>Präsenzzeit Modul insgesamt</b>				<b>84 h</b>

---

---

## inf031 - Objektorientierte Modellierung und Programmierung

<b>Modulbezeichnung</b>	Objektorientierte Modellierung und Programmierung
<b>Modulkürzel</b>	inf031
<b>Kreditpunkte</b>	9.0 KP
<b>Workload</b>	270 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"><li>• Fach-Bachelor Informatik (Bachelor) &gt; Basismodule</li><li>• Fach-Bachelor Mathematik (Bachelor) &gt; Nebenfachmodule</li><li>• Fach-Bachelor Wirtschaftsinformatik (Bachelor) &gt; Basiscurriculum</li><li>• Fach-Bachelor Wirtschaftswissenschaften (Bachelor) &gt; Studienrichtung Wirtschaftsinformatik</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Basismodule</li></ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"><li>• Schönberg, Christian (Modulverantwortung)</li><li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li></ul>
<b>Teilnahmevoraussetzungen</b>	

### Nützliche fachspezifische Vorkenntnisse:

- Imperative Programmierung mit Java
- Grundlagen der Objektorientierung (Klassen und Objekte)
- Grundlagen der Algorithmik (Komplexität, Lösungsmethoden)

Diese Vorkenntnisse können **beispielsweise** im Modul *inf030 Programmierung, Datenstrukturen und Algorithmen* erworben werden. Formale Teilnahmevoraussetzungen gibt es **nicht**.

---

### Kompetenzziele

Die Objektorientierung stellt heutzutage den Stand der Technik in der Softwareentwicklung dar. Gegebene Problemstellungen werden dabei mit Hilfe objektorientierter Analyse- und Entwurfsverfahren zunächst in ein objektorientiertes Modell und anschließend in ein objektorientiertes Programm überführt. Ziel des Moduls „Objektorientierte Modellierung und Programmierung“ ist das Erlernen grundlegender Konzepte der objektorientierten Modellierung mit Hilfe der UML als Modellierungsnotation und der objektorientierten Programmierung mit der Programmiersprache Java. Die Studierenden sollen nach Abschluss des Moduls selbstständig objektorientierte Programme auf der Grundlage von Java für die Lösung mittelgroßer Probleme entwickeln können.

### Fachkompetenzen

Die Studierenden

- kennen grundlegende Konzepte der objektorientierten Modellierung und UML als Modellierungsnotation
- kennen grundlegende Konzepte der objektorientierten Programmierung mit Java
- kennen die Terminologie der objektorientierten Modellierung und Programmierung und verwenden die entsprechenden Begriffe präzise bei Diskussionen
- können beschreiben, was ihnen vorgelegte objektorientierte Programme tun
- entwickeln selbstständig Modelle und Programme für die Lösung mittelgroßer Probleme
- untersuchen systematisch eigene und fremde Modelle und Programme auf Fehler
- setzen moderne Entwicklungsumgebungen zum Modellieren und Entwickeln von Programmen ein
- kennen die Unterschiede zwischen dem imperativen, objektorientierten, funktionalen, logischen und regelbasierten Programmierparadigma

### Methodenkompetenzen

Die Studierenden

- entwickeln selbstständig Programme für gegebene Probleme durch konsequente Anwendung der Konzepte der objektorientierten Modellierung und Programmierung

- 
- übertragen praktische Erfahrungen in der Programmentwicklung auf neue Aufgaben
  - entwickeln selbstständig Programme mit Nebenläufigkeiten
  - können selbstständig bekannte Lösungsmethoden auf komplexe Probleme anwenden

### **Sozialkompetenzen**

Die Studierenden

- vermitteln die Struktur und Wirkungsweise selbst entwickelter Modelle und Programme an andere
- präsentieren selbstständig entwickelte Lösungen zu kleinen Aufgaben vor Gruppen

### **Selbstkompetenzen**

Die Studierenden

- organisieren sich beim Entwickeln von Modellen und Programmen für kleine und mittelgroße Probleme der Informatik
- beziehen die Konzepte des objektorientierten Programmentwurfs in ihr Handeln ein

---

## **Modulinhalte**

Im ersten Teil werden grundlegende Konzepte der objektorientierten Modellierung und Programmierung vermittelt:

- Modelle und Modellierung
- UML-Klassendiagramme
- Klassen und Objekte
- Datenkapselung
- Vererbung
- Polymorphie und dynamisches Binden
- Ausnahmebehandlung
- Generizität

Im zweiten Teil werden wichtige Konzepte und Klassen der JDK-Klassenbibliothek vorgestellt und die Klassen bei der Lösung mittelgroßer Probleme eingesetzt:

- Java-Collection-API
- I/O und Streams
- Parallele Programmierung mit Threads
- GUI-Anwendungen mit JavaFX

Im dritten Teil werden fortgeschrittene Lösungsstrategien vorgestellt sowie weitere Programmierparadigmen eingeführt und mit dem objektorientierten Paradigma verglichen:

- Backtracking, Branch and Bound, Greedy
- Lokale Suche, Evolutionäre Algorithmen
- Funktionale Programmierung (z.B. Java-Lamdas, Standard ML)
- Logische Programmierung (z.B. Prolog)
- Regelbasierte Programmierung (z.B. RuleBook)

Ergänzt wird der Vorlesungsteil um einen umfassenden Übungsteil, in dem insbesondere die vermittelten Inhalte an praktischen Beispielen umgesetzt werden.

---

## **Literaturempfehlungen**

### **Essenziell**

- Skript (wird in elektronischer Form über das Stud.IP fortlaufend in der Vorlesung zur Verfügung gestellt)

### **Empfohlene Sekundärliteratur**



- Heide Balzert: Lehrbuch der Objektmodellierung: Analyse und Entwurf mit der UML 2, Spektrum Akademischer Verlag
- Dietmar Ratz, Jens Scheffler, Detlev Seese, Jan Wiesenberger: Grundkurs Programmieren in Java, Carl Hanser Verlag.
- Christian Ullenboom: Java ist auch eine Insel: Programmieren lernen mit dem Standardwerk für Java- Entwickler, Rheinwerk Computing
- Christian Ullenboom: Java SE 8 Standard-Bibliothek: Das Handbuch für Entwickler, Rheinwerk Computing

<b>Links</b>				
<b>Unterrichtssprache</b>		Deutsch		
<b>Dauer in Semestern</b>		1 Semester		
<b>Angebotsrhythmus Modul</b>		jedes Sommersemester		
<b>Aufnahmekapazität Modul</b>		unbegrenzt		
<b>Lehr-/Lernform</b>		1V + 1Ü		
<b>Vorkenntnisse</b>		Nützliche Vorkenntnisse: inf030 Programmierung, Datenstrukturen und Algorithmen		
Prüfung	Prüfungszeiten	Prüfungsform		
<b>Gesamtmodul</b>		Am Ende der Veranstaltung	Klausur / Portfolio (schriftliche Kurzprüfungen) / mündliche Prüfung	
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenz
Vorlesung		4	SoSe	56
Übung		2	SoSe	28
<b>Präsenzzeit Modul insgesamt</b>				84 h

---

## inf200 - Grundlagen der Technischen Informatik

<b>Modulbezeichnung</b>	Grundlagen der Technischen Informatik
<b>Modulkürzel</b>	inf200
<b>Kreditpunkte</b>	6.0 KP
<b>Workload</b>	180 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"><li>• Fach-Bachelor Informatik (Bachelor) &gt; Basismodule</li><li>• Fach-Bachelor Mathematik (Bachelor) &gt; Nebenfachmodule</li><li>• Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) &gt; Akzentsetzungsbereich</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Basismodule</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Wahlpflicht Technische Informatik (30 KP)</li></ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"><li>• Rauh, Andreas (Modulverantwortung)</li><li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li></ul>

### Teilnahmevoraussetzungen

Keine Teilnehmvoraussetzungen

### Kompetenzziele

Die Studierenden verstehen den Aufbau digitaler Schaltkreise und Rechnersysteme und verfügen über Kenntnisse der grundlegenden technologischen Parameter, Kriterien, Voraussetzungen und Entwicklungen des derzeitigen und zukünftig zu erwartenden Entwurfs digitaler Hardware. Sie verstehen die Grundkonzepte aktueller Rechnerarchitekturen und des Ablaufs von Programmen hierauf. Am Ende der Veranstaltung sollen die Studierenden in der Lage sein, Rechnerarchitekturen zu analysieren, einzelne Hardwarekomponenten von Rechnern zu verstehen, sie zu entwerfen und zu optimieren sowie qualifiziert die Eigenschaften grundlegender Entwurfsalternativen zu diskutieren.

#### Fachkompetenzen

Die Studierenden:

- identifizieren grundlegende Konzepte des Aufbaus digitaler Rechnersysteme, der internen Zahlendarstellung und Schatnetzanalyse sowie deren Optimierung

#### Methodenkompetenzen

Die Studierenden:

- analysieren Rechnerarchitekturen anhand einzelner Komponenten
- entwerfen und optimieren einzelne Komponenten von Rechnern
- transferieren systematische Methoden des Schaltkreisentwurfs auf neue Problemstellungen

#### Sozialkompetenzen

Die Studierenden:

- präsentieren ihr Verständnis der Funktionsprinzipien digitaler Rechnersysteme

#### Selbstkompetenzen

Die Studierenden:

- reflektieren die Ergebnisse von Übungsaufgaben kritisch und erkennen Grenzen unterschiedlicher Ansätze für den Entwurf digitaler Rechnersysteme

### Modulinhalte

Dieses Modul ist der erste Teil der zweisemestrigen Einführung in die Technische Informatik. Es erläutert die Konstruktionsprinzipien eines Rechners von der Ausführung eines einfachen Programms auf einer Instruction Set Architecture über die grundlegenden Techniken zur Kodierung und Zahlendarstellung, dem Programmablauf auf Maschinenebene, Grundlagen der Logik und Schatnetzanalyse sowie deren Optimierung.

## Literaturempfehlungen

- Skript zur Vorlesung
- Schiffmann, W.; Schmitz, R. (2001): Technische Informatik I, II, Übungsbuch; Springer Verlag, Berlin.
- Dal Cin, M. (1996): Rechnerarchitektur; B.G. Teubner.
- Lagemann, K. (1987): Rechnerstrukturen; Springer-Verlag, Berlin.
- Oberschelp, W.; Vossen, G. (1989): Rechneraufbau und Rechnerstrukturen; Oldenbourg-Verlag.
- Mano, Morris M. (1993): Computer System Architecture 3; Prentice Hall.
- Gajski, D. (1997): Principles of Digital Design; Prentice Hall.
- Patterson, D.A.; Hennessy, J.L. (1997): Computer Organization and Design:
- The Hardware/Software Interface; 2. Edition; Morgan Kaufmann Publishers.
- Wilkinson, B. (1996): Computer Architecture Design and Performance; 2. Edition; Prentice Hall.
- Tanenbaum, A.S. (1999): Structured Computer Organization; 4. Edition; Prentice Hall.

Links				
<b>Unterrichtssprache</b>	Deutsch			
<b>Dauer in Semestern</b>	1 Semester			
<b>Angebotsrhythmus Modul</b>	jährlich			
<b>Aufnahmekapazität Modul</b>	unbegrenzt			
<b>Lehr-/Lernform</b>	1VL + 1Ü			
<b>Vorkenntnisse</b>	keine			
Prüfung	Prüfungszeiten	Prüfungsform		
<b>Gesamtmodul</b>	Am Ende der Vorlesungszeit	Klausur oder mündliche Prüfung		
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenz
Vorlesung		3	WiSe	42
Übung		1	WiSe	14
<b>Präsenzzeit Modul insgesamt</b>				<b>56 h</b>

---

## inf700 - Didaktik der Informatik I

<b>Modulbezeichnung</b>	Didaktik der Informatik I
<b>Modulkürzel</b>	inf700
<b>Kreditpunkte</b>	6.0 KP
<b>Workload</b>	180 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"><li>• Fach-Bachelor Informatik (Bachelor) &gt; Akzentsetzungsbereich - Wahlbereich Informatik</li><li>• Fach-Bachelor Informatik (Bachelor) &gt; Wahlbereich Informatik, Kultur und Gesellschaft</li><li>• Fach-Bachelor Wirtschaftsinformatik (Bachelor) &gt; Wahlbereich Informatik, Kultur und Gesellschaft</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Aufbaumodule (60 KP)</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Basismodule</li></ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"><li>• Diethelm, Ira (Modulverantwortung)</li><li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li></ul>

### Teilnahmevoraussetzungen

Fachliche Grundkenntnisse der Informatik

---

### Kompetenzziele

#### Fachkompetenz

Die Studierenden:

- charakterisieren die erlernten Konzepte und Ansätze zur Didaktik der Informatik, wie z.B. die frühen Ansätze der Schulinformatik oder das Konzept zur Informatik im Kontext
- differenzieren und diskutieren didaktische Ansätze und Konzepte zur Auswahl von informatischen Inhalten für den Schulunterricht
- argumentieren den allgemeinbildenden Charakter der Informatik und vergleichen die erlernten Ansätze und Konzepte zur Didaktik der Informatik und illustrieren Gemeinsamkeiten und Widersprüche
- sind in der Lage Themen für den Informatik-Unterricht anhand der erlernten Konzepte und Ansätze zur Didaktik der Informatik zu reflektieren

#### Methodenkompetenz

Die Studierenden:

- vernetzen die Konzepte und Ansätze der Didaktik der Informatik mit Hilfe der didaktischen Rekonstruktion
- klassifizieren die Gemeinsamkeiten und Unterschiede der Ansätze und Konzepte der Didaktik der Informatik

#### Sozialkompetenz

Die Studierenden:

- diskutieren die Ansätze und Konzepte der Didaktik der Informatik mit Kommilitonen
- akzeptieren Meinungen anderer und nehmen sachliche Kritik an
- äußern konstruktive Kritik

#### Selbstkompetenz

Die Studierenden:

- beziehen die Ansätze und Konzepte der Didaktik der Informatik in ihr Handeln ein
- reflektieren ihr Selbstbild unter den Gesichtspunkten der Ansätze und Konzept der Didaktik der Informatik

---

### Modulinhalte

In der Veranstaltung wird in das Fachgebiet Didaktik der Informatik eingeführt. Dabei werden verschiedene Konzepte und Ansätze zur Didaktik der Informatik vorgestellt.

#### Inhalte sind:

- frühe Konzepte des Informatik-Unterrichts
- Allgemeinbildung und Informatik-Unterricht

- der Ideenorientierte Ansatz
- der Informationszentrierte Ansatz
- Grundschulinformatik
- der Systemorientierte Ansatz Darüber hinaus werden zentrale Themen, wie zum Beispiel: "Projekte im Informatikunterricht", aufgegriffen.

#### Literaturempfehlungen

- Schwill, A.; Schubert, S.: Didaktik der Informatik. Berlin: Spektrum Akademischer Verlag, 2004
- Hubwieser, P.: Didaktik der Informatik. Berlin: Springer Verlag, 2000

#### Links

<b>Unterrichtssprache</b>	Deutsch			
<b>Dauer in Semestern</b>	1 Semester			
<b>Angebotsrhythmus Modul</b>	jährlich			
<b>Aufnahmekapazität Modul</b>	unbegrenzt			
<b>Lehr-/Lernform</b>	V+Ü			
<b>Vorkenntnisse</b>	Fachliche Grundkenntnisse der Informatik			
<b>Prüfung</b>	<b>Prüfungszeiten</b>	<b>Prüfungsform</b>		
<b>Gesamtmodul</b>	Am Ende der Vorlesungszeit	Mündliche Prüfung		
<b>Lehrveranstaltungsform</b>	<b>Kommentar</b>	<b>SWS</b>	<b>Angebotsrhythmus</b>	<b>Workload Präsenz</b>
Vorlesung		2	SoSe	28
Übung		2	SoSe	28
<b>Präsenzzeit Modul insgesamt</b>				<b>56 h</b>

## mat950 - Mathematik für Informatik (Diskrete Strukturen)

<b>Modulbezeichnung</b>	Mathematik für Informatik (Diskrete Strukturen)			
<b>Modulkürzel</b>	mat950			
<b>Kreditpunkte</b>	6.0 KP			
<b>Workload</b>	180 h			
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"> <li>• Fach-Bachelor Informatik (Bachelor) &gt; Aufbaumodule</li> <li>• Fach-Bachelor Wirtschaftsinformatik (Bachelor) &gt; Aufbaucurriculum - Pflichtbereich</li> <li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Basismodule</li> </ul>			
<b>Zuständige Personen</b>	<ul style="list-style-type: none"> <li>• Heß, Florian (Modulverantwortung)</li> <li>• Stein, Andreas (Modulverantwortung)</li> <li>• Stein, Sandra (Modulverantwortung)</li> </ul>			
<b>Teilnahmevoraussetzungen</b>				
<b>Kompetenzziele</b>	<p>- Kennenlernen und Verstehen des axiomatischen Aufbaus der Mathematik und der Bedeutung mathematischer Argumentation - Beherrschen grundlegender mathematischer Beweistechniken und deren logischer Struktur - Erkennen der Bedeutung von Voraussetzungen in mathematischen Sätzen: Lokalisierung der Voraussetzungen innerhalb der Beweise und mögliche Konsequenzen bei Wegfall von Voraussetzungen - Exemplarisches Kennenlernen weiterer mathematischer Gebiete und damit Erweiterung des eigenen mathematischen Wissens - Kennenlernen von Anwendungen - Vernetzung des eigenen mathematischen Wissens durch Herstellung von Bezügen zwischen verschiedenen mathematischen Bereichen - Erlernen der wesentlichen Ideen und Methoden von diskreten Strukturen in der Mathematik - Beherrschen der Grundbegriffe und wesentlichen Methoden der Graphentheorie - Beherrschen der Grundbegriffe und wesentlichen Methoden der Algebra und Zahlentheorie, wie Gruppen, Ringe, Körper, Restklassenringe, euklidischer Algorithmus, chinesischer Restsatz, Polynome - Beherrschen weiterführender Begriffe und Methoden für diskrete Strukturen, wie z.B. Primzahltests, RSA, graphentheoretische Algorithmen</p>			
<b>Modulinhalte</b>	Elemente der Aussagenlogik, Beweismethoden, Mengen, Relationen und Abbildungen, Kombinatorik, Graphen und Anwendungen, die ganzen Zahlen und ihre Restklassenringe, Gruppen und Halbgruppen			
<b>Literaturempfehlungen</b>	B. Kreußler und G. Pfister: Mathematik für Informatiker, Springer-Verlag 2009 (campusweiter Online-Zugriff auf den Volltext über das Bibliothekssystem) Weitere Literatur wird in der Veranstaltung bekannt gegeben.			
<b>Links</b>				
<b>Unterrichtssprache</b>	Deutsch			
<b>Dauer in Semestern</b>	1 Semester			
<b>Angebotsrhythmus Modul</b>	jährlich			
<b>Aufnahmekapazität Modul</b>	unbegrenzt			
<b>Hinweise</b>	Im Zwei-Fächer Bachelor Informatik ist dieses Modul im Basiscurriculum zu studieren.			
<b>Modulart</b>	Pflicht / Mandatory			
<b>Modullevel</b>	AC (Aufbaucurriculum / Composition)			
<b>Lehr-/Lernform</b>	Vorlesung + Übung			
Prüfung	Prüfungszeiten		Prüfungsform	
<b>Gesamtmodul</b>	Klausur nach Abschluss der Vorlesung		Klausur oder mündliche Prüfung	
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenz
Vorlesung		3	WiSe	42
Übung		1	WiSe	14
<b>Präsenzzeit Modul insgesamt</b>				56 h

---

## Aufbaumodule (60 KP)

### inf005 - Softwaretechnik I

<b>Modulbezeichnung</b>	Softwaretechnik I
<b>Modulkürzel</b>	inf005
<b>Kreditpunkte</b>	6.0 KP
<b>Workload</b>	180 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"><li>• Fach-Bachelor Informatik (Bachelor) &gt; Aufbaumodule</li><li>• Fach-Bachelor Mathematik (Bachelor) &gt; Nebenfachmodule</li><li>• Fach-Bachelor Wirtschaftsinformatik (Bachelor) &gt; Aufbaucurriculum - Pflichtbereich</li><li>• Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) &gt; Pflichtbereich</li><li>• Master Umweltmodellierung (Master) &gt; Mastermodule</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Aufbaumodule (60 KP)</li></ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"><li>• Winter, Andreas (Modulverantwortung)</li><li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li></ul>
<b>Teilnahmevoraussetzungen</b>	

## Erwartete/Nützliche Vorkenntnisse

### aus inf030 **Programmierung, Datenstrukturen und Algorithmen**

#### Fachkompetenzen

Die Studierenden:

- beschreiben grundlegende Konzepte der imperativen Programmierung mit Java
- erkennen die Terminologie der imperativen Programmierung und verwenden die entsprechenden Begriffe präzise bei Diskussionen
- erkennen grundlegende Terminologie der objektorientierten Programmierung
- beschreiben, was ihnen vorgelegte Programme tun
- entwickeln selbstständig Programme für die Lösung kleinerer Probleme
- untersuchen systematisch eigene und fremde Programme auf Fehler
- setzen moderne Programmierungsumgebungen zum Entwickeln und Testen von Programmen ein
- erstellen Algorithmen mit allgemeinen Entwurfskonzepten (z.B. Greedy-Verfahren, Divide-and-Conquer-Verfahren)
- benennen Algorithmen und Datenstrukturen zur Lösung von häufig vorkommenden Problemen und bewerten diese in ihrer Anwendbarkeit
- benennen Probleme der Effizienz von algorithmischen Lösungen konkreter Fragestellungen und bewerten diese
- wählen fundiert einen Algorithmus und eine Datenstruktur zur Lösung eines konkreten Problems aus
- wenden die gelernten Algorithmen und Datenstrukturen sinnvoll auf gegebene und konkrete Probleme an

#### Methodenkompetenzen

Die Studierende:

- lösen gegebene Probleme unter den Gesichtspunkt der imperativen bzw. objektorientierten Programmierung
- übertragen praktische Erfahrungen in der Programmentwicklung auf neue Aufgaben

#### Sozialkompetenzen

Die Studierenden

- vermitteln die Struktur und Wirkungsweise selbst entwickelter Programme an andere
- präsentieren Lösungen zu kleinen Aufgaben vor Gruppen

#### Selbstkompetenzen

Die Studierenden:

- organisieren sich beim Finden von algorithmischen Lösungen für kleine und mittelgroße Probleme der Informatik

- 
- beziehen die Konzepte des allgemeinen Programmentwurfs in ihr Handeln ein

## aus inf031 Objektorientierte Modellierung und Programmierung

### Fachkompetenzen

Die Studierendende:

- kennen grundlegende Konzepte der objektorientierten Modellierung und UML als Modellierungsnotation
- kennen grundlegende Konzepte der objektorientierten Programmierung mit Java
- kennen die Terminologie der objektorientierten Modellierung und Programmierung und verwenden die entsprechenden Begriffe präzise bei Diskussionen
- können beschreiben, was ihnen vorgelegte objektorientierte Programme tun
- entwickeln selbstständig Modelle und Programme für die Lösung mittelgroßer Probleme
- untersuchen systematisch eigene und fremde Modelle und Programme auf Fehler
- setzen moderne Entwicklungsumgebungen zum Modellieren und Entwickeln von Programmen ein
- kennen die Unterschiede zwischen dem imperativen, objektorientierten, funktionalen, logischen und regelbasierten Programmierparadigma

### Methodenkompetenzen

Die Studierenden:

- entwickeln selbstständig Programme für gegebene Probleme durch konsequente Anwendung der Konzepte der objektorientierten Modellierung und Programmierung
- übertragen praktische Erfahrungen in der Programmentwicklung auf neue Aufgaben
- entwickeln selbstständig Programme mit Nebenläufigkeiten
- können selbstständig bekannte Lösungsmethoden auf komplexe Probleme anwenden

### Sozialkompetenzen

Die Studierenden:

- vermitteln die Struktur und Wirkungsweise selbst entwickelter Modelle und Programme an andere
- präsentieren selbstständig entwickelte Lösungen vor Gruppen

### Selbstkompetenzen

Die Studierenden:

- organisieren sich beim Entwickeln von Programmen für kleine und mittelgroße Probleme der Informatik
- beziehen die Konzepte des objektorientierten Programmentwurfs in ihr Handeln ein

---

### Kompetenzziele

Ziel des Moduls ist die Vermittlung der ingenieurmäßigen Entwicklung und Wartung umfangreicher Softwaresysteme. Betrachtet wird der vollständige Software-Entwicklungsprozess inkl. Anforderungserhebung, Software-Architektur und Qualitätssicherung sowohl in klassischen wie in agilen Vorgehensweisen. Vertieft werden Grundkonzepte der objektorientierten Modellierung und Softwareentwicklung auf Basis der Unified Modeling Language.

### Fachkompetenzen

Die Studierenden:

- erkennen die Phasen im Software-Lebenszyklus (Anforderungsermittlung, Entwurf, Implementierung, Qualitätssicherung)
- benennen die in den Phasen anfallenden Aufgaben
- erkennen und bewerten die Anordnung dieser Phasen in klassischen und agilen Vorgehensweisen
- beurteilen und wählen geeignete Vorgehensweisen zur Umsetzung von



- Projekten aus
- erkennen die Sprachmöglichkeiten der Modellierung mit UML
- entwickeln und bewerten Modelle in unterschiedlichen UML-Notationen und deren Kombinationen
- lösen gegebene Probleme mit Hilfe der UML-Notationen

#### **Methodenkompetenzen**

Die Studierenden:

- strukturieren, bewerten, unterscheiden und nutzen Vorgehenweisen der klassischen und agilen Projektdurchführung
- strukturieren, dokumentieren, bewerten Probleme und Lösungen mit den Werkzeugen der objekt-orientierten Modellierung
- wenden Methoden und Techniken der objekt-orientierten Modellierung mit UML gezielt an

#### **Sozialkompetenzen**

Die Studierenden:

- erstellen, präsentieren und diskutieren Problemlösungen mit Hilfe von Modellierungstechniken
- beschreiben und lösen gegebenen Probleme der Modellierung in Gruppen

#### **Selbstkompetenzen**

Die Studierenden:

- reflektieren ihr Handeln bei der Problembeschreibung und der Entwicklung von Lösungsansätzen

---

#### **Modulinhalte**

In dem Modul werden die grundlegenden Begriffe und Konzepte der Softwaretechnik vermittelt.

Es sind dies u.a.:

- Notwendigkeit der Softwaretechnik
- Prinzipien der Softwaretechnik
- Aktivitäten und Vorgehensmodelle der Software-Entwicklung (klassisch, agil)
- Objektorientierte Modellierung, Metamodellierung
- Synchronisation von Code und Modellen
- Ermittlung und Dokumentation von Anforderung (klassisch, agil)
- Definition von Software-Architekturen
- Einsatz von Mustern der Software Entwicklung
- Definition und Sicherung der Softwarequalität
- Wartung und Betrieb von Softwaresystemen

---

#### **Literaturempfehlungen**

- Folienskript zur Vorlesung
- Ian Sommerville: Software Engineering, Addison-Wesley Longman, Amsterdam, 10. Auflage (Global Edition). 2015.
- Helmut Balzert: Lehrbuch der Software-Technik, Spektrum Akademischer Verlag, 3. Auflage 2009.
- Anja Metzner: Software-Engineering – kompakt, Hanser, München, 2020.
- Ravi Sethi: Software Engineering: Basic Principles and Best Practices, Cambridge University Press, 8. Dezember 2022.
- Chris Rupp, Stefan Queins: UML 2 glasklar. Praxiswissen für die UML-Modellierung, Carl Hanser Verlag, 4. Auflage 2012.
- Martina Seidl, Marion Scholz, Christian Huemer, Gerti Kappel, UML @ Classroom: An Introduction to Object-Oriented Modeling, Springer, 2015.
- Christoph Kecher, Alexander Salvanos, Ralf Hoffmann-Elbern: UML 2.5, Das umfassende Handbuch. 7. Aufl. Rheinwerk Computing, 2021.
- OMG Unified Modeling Language, Version 2.5.1 (formal/17-12-05), Dec. 2017, <https://www.omg.org/spec/UML/>,

---

#### **Links**

<b>Unterrichtssprache</b>	Deutsch
<b>Dauer in Semestern</b>	1 Semester
<b>Angebotsrhythmus Modul</b>	jährlich

<b>Aufnahmekapazität Modul</b>	unbegrenzt
<b>Lehr-/Lernform</b>	V+Ü
<b>Vorkenntnisse</b>	- inf030 - inf031

Prüfung	Prüfungszeiten	Prüfungsform
<b>Gesamtmodul</b>	Am Ende der Vorlesungszeit	Klausur (im Regelfall)  mündliche Prüfung oder Portfolio (nach Absprache mit dem Prüfungsamt z.B. bei bewilligtem Nachteilsausgleich)

Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenz
Vorlesung		3	WiSe	42
Übung		2	WiSe	28
<b>Präsenzzeit Modul insgesamt</b>				<b>70 h</b>

---

## inf007 - Informationssysteme I

<b>Modulbezeichnung</b>	Informationssysteme I
<b>Modulkürzel</b>	inf007
<b>Kreditpunkte</b>	6.0 KP
<b>Workload</b>	180 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"><li>• Fach-Bachelor Informatik (Bachelor) &gt; Aufbaumodule</li><li>• Fach-Bachelor Wirtschaftsinformatik (Bachelor) &gt; Aufbaucurriculum - Pflichtbereich</li><li>• Fach-Bachelor Wirtschaftswissenschaften (Bachelor) &gt; Studienrichtung Wirtschaftsinformatik</li><li>• Master Applied Economics and Data Science (Master) &gt; Specialization</li><li>• Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) &gt; Pflichtbereich</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Aufbaumodule (60 KP)</li></ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"><li>• Wingerath, Wolfram (Modulverantwortung)</li><li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li></ul>
<b>Teilnahmevoraussetzungen</b>	Keine Teilnehmvoraussetzungen
<b>Kompetenzziele</b>	<p>Dies Modul behandelt grundlegende Konzepte, Sprachen und Architekturen von Datenbanken (DB), die einen wichtigen Baustein zur Realisierung moderner Softwaresysteme darstellen.</p> <p><b>Fachkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• verfügen über Kenntnisse zu grundlegenden Konzepten, Sprachen und Architekturen von (insbesondere relationalen) Datenbanken</li><li>• wählen Datenmodelle begründet aus</li><li>• integrieren weitergehende Konzepte von Informationssystemen in ihre Überlegungen</li></ul> <p><b>Methodenkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• entwerfen Datenbanksysteme in sinnvollen Zusammenhängen</li><li>• analysieren Probleme aus dem Bereich der datenbankgestützten Informationsverarbeitung methodisch und schlagen Lösungen vor</li></ul> <p><b>Sozialkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• vertiefen ihre Fähigkeit zur Arbeit im Team</li></ul> <p><b>Selbstkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• reflektieren ihr Handeln beim Identifizieren von Lösungsansätzen und beziehen dabei die Konzepte der Informationsverarbeitung ein</li></ul>
<b>Modulinhalte</b>	<ul style="list-style-type: none"><li>• Relationales Datenmodell - Relationalenalgebra und deren Implementierung in SQL (dem Sprachstandard für Datenbanken)</li><li>• Entwurf von Datenbanken auf unterschiedlichen Abstraktionsebenen (konzeptionelles und logisches Design)</li><li>• Normalformen</li><li>• Datenbank-Architekturen</li><li>• verteilte und aktive Datenbanken sowie objektorientierte, objektrelationale und XML-basierte Datenbank-Systeme</li></ul>
<b>Literaturempfehlungen</b>	<ul style="list-style-type: none"><li>• Ramez Elmasri und Shamkant B. Navathe (2016) - Fundamentals of Databases Systems. Seventh Edition, Pearson/Addison Wesley</li></ul>
<b>Links</b>	

<b>Unterrichtssprache</b>	Deutsch			
<b>Dauer in Semestern</b>	1 Semester			
<b>Angebotsrhythmus Modul</b>	jährlich			
<b>Aufnahmekapazität Modul</b>	unbegrenzt			
<b>Lehr-/Lernform</b>	V+Ü			
<b>Vorkenntnisse</b>	keine			
<b>Prüfung</b>	<b>Prüfungszeiten</b>	<b>Prüfungsform</b>		
<b>Gesamtmodul</b>	Am Ende der Vorlesungszeit		Klausur oder mündliche Prüfung	
<b>Lehrveranstaltungsform</b>	<b>Kommentar</b>	<b>SWS</b>	<b>Angebotsrhythmus</b>	<b>Workload Präsenz</b>
Vorlesung		3	WiSe	42
Übung		1	WiSe	14
<b>Präsenzzeit Modul insgesamt</b>				<b>56 h</b>

---

## inf700 - Didaktik der Informatik I

<b>Modulbezeichnung</b>	Didaktik der Informatik I
<b>Modulkürzel</b>	inf700
<b>Kreditpunkte</b>	6.0 KP
<b>Workload</b>	180 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"><li>• Fach-Bachelor Informatik (Bachelor) &gt; Akzentsetzungsbereich - Wahlbereich Informatik</li><li>• Fach-Bachelor Informatik (Bachelor) &gt; Wahlbereich Informatik, Kultur und Gesellschaft</li><li>• Fach-Bachelor Wirtschaftsinformatik (Bachelor) &gt; Wahlbereich Informatik, Kultur und Gesellschaft</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Aufbaumodule (60 KP)</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Basismodule</li></ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"><li>• Diethelm, Ira (Modulverantwortung)</li><li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li></ul>

### Teilnahmevoraussetzungen

Fachliche Grundkenntnisse der Informatik

---

### Kompetenzziele

#### Fachkompetenz

Die Studierenden:

- charakterisieren die erlernten Konzepte und Ansätze zur Didaktik der Informatik, wie z.B. die frühen Ansätze der Schulinformatik oder das Konzept zur Informatik im Kontext
- differenzieren und diskutieren didaktische Ansätze und Konzepte zur Auswahl von informatischen Inhalten für den Schulunterricht
- argumentieren den allgemeinbildenden Charakter der Informatik und vergleichen die erlernten Ansätze und Konzepte zur Didaktik der Informatik und illustrieren Gemeinsamkeiten und Widersprüche
- sind in der Lage Themen für den Informatik-Unterricht anhand der erlernten Konzepte und Ansätze zur Didaktik der Informatik zu reflektieren

#### Methodenkompetenz

Die Studierenden:

- vernetzen die Konzepte und Ansätze der Didaktik der Informatik mit Hilfe der didaktischen Rekonstruktion
- klassifizieren die Gemeinsamkeiten und Unterschiede der Ansätze und Konzepte der Didaktik der Informatik

#### Sozialkompetenz

Die Studierenden:

- diskutieren die Ansätze und Konzepte der Didaktik der Informatik mit Kommilitonen
- akzeptieren Meinungen anderer und nehmen sachliche Kritik an
- äußern konstruktive Kritik

#### Selbstkompetenz

Die Studierenden:

- beziehen die Ansätze und Konzepte der Didaktik der Informatik in ihr Handeln ein
- reflektieren ihr Selbstbild unter den Gesichtspunkten der Ansätze und Konzept der Didaktik der Informatik

---

### Modulinhalte

In der Veranstaltung wird in das Fachgebiet Didaktik der Informatik eingeführt. Dabei werden verschiedene Konzepte und Ansätze zur Didaktik der Informatik vorgestellt.

#### Inhalte sind:

- frühe Konzepte des Informatik-Unterrichts
- Allgemeinbildung und Informatik-Unterricht

- der Ideenorientierte Ansatz
- der Informationszentrierte Ansatz
- Grundschulinformatik
- der Systemorientierte Ansatz Darüber hinaus werden zentrale Themen, wie zum Beispiel: "Projekte im Informatikunterricht", aufgegriffen.

#### Literaturempfehlungen

- Schwill, A.; Schubert, S.: Didaktik der Informatik. Berlin: Spektrum Akademischer Verlag, 2004
- Hubwieser, P.: Didaktik der Informatik. Berlin: Springer Verlag, 2000

#### Links

<b>Unterrichtssprache</b>	Deutsch			
<b>Dauer in Semestern</b>	1 Semester			
<b>Angebotsrhythmus Modul</b>	jährlich			
<b>Aufnahmekapazität Modul</b>	unbegrenzt			
<b>Lehr-/Lernform</b>	V+Ü			
<b>Vorkenntnisse</b>	Fachliche Grundkenntnisse der Informatik			
<b>Prüfung</b>	<b>Prüfungszeiten</b>	<b>Prüfungsform</b>		
<b>Gesamtmodul</b>	Am Ende der Vorlesungszeit	Mündliche Prüfung		
<b>Lehrveranstaltungsform</b>	<b>Kommentar</b>	<b>SWS</b>	<b>Angebotsrhythmus</b>	<b>Workload Präsenz</b>
Vorlesung		2	SoSe	28
Übung		2	SoSe	28
<b>Präsenzzeit Modul insgesamt</b>				<b>56 h</b>

---

## Praktische Vertiefung (60 KP)

### inf004 - Softwareprojekt

<b>Modulbezeichnung</b>	Softwareprojekt
<b>Modulkürzel</b>	inf004
<b>Kreditpunkte</b>	9.0 KP
<b>Workload</b>	270 h
<b>Verwendbarkeit des Moduls</b>	

- Fach-Bachelor Betriebswirtschaftslehre mit juristischem Schwerpunkt (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Biologie (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Chemie (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Comparative and European Law (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Engineering Physics (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel mehr...
- Fach-Bachelor Informatik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Interkulturelle Bildung und Beratung (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Mathematik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Nachhaltigkeitsökonomik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Pädagogik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Pädagogisches Handeln in der Migrationsgesellschaft (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Physik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Physik, Technik und Medizin (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Sozialwissenschaften (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Umweltwissenschaften (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Wirtschaftsinformatik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Wirtschaftswissenschaften (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Anglistik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Biologie (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Chemie (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Elementarmathematik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Ev. Theologie und Religionspädagogik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Gender Studies (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Germanistik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Geschichte (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Informatik (Bachelor) > Praktische Vertiefung (60 KP)
- Zwei-Fächer-Bachelor Informatik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Interdisziplinäre Sachbildung (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Kunst und Medien (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Materielle Kultur: Textil (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel

- Zwei-Fächer-Bachelor Mathematik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Musik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Niederdeutsch (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Niederlandistik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Ökonomische Bildung (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Pädagogik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Philosophie / Werte u. Normen (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Physik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Politik-Wirtschaft (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Slavistik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Sonderpädagogik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Sozialwissenschaften (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Sportwissenschaft (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Technik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Wirtschaftswissenschaften (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel

---

#### Zuständige Personen

- Grawunder, Marco (Modulverantwortung)
- Wüstefeld, Manuela (Modulverantwortung)
- Lehrenden, Die im Modul (Prüfungsberechtigt)

---

#### Teilnahmevoraussetzungen

##### Nützliche Vorkenntnisse:

- PDA
- OMP

---

#### Kompetenzziele

Die Studierenden verfügen über praktische Kenntnisse zur Software-Entwicklung in einem iterativen Vorgehen im Team und haben dabei die wesentlichen Phasen des Software-Lebenszyklus (Anforderungen, Analyse, Entwurf, Implementierung, Test) betrachtet und regelmäßig ihre Ergebnisse präsentiert. Sie haben ihre Kenntnisse in Java vertieft.

##### Fachkompetenzen

Die Studierenden:

- wenden Techniken und Methoden an und erkennen ihre Grenzen

##### Methodenkompetenzen

Die Studierenden:

- entwickeln komplexere Software mit Methoden des Software-Engineering anhand eines Vorgehensmodells und dokumentieren diese entsprechend
- schätzen kleinere Aufgaben grob ab
- führen einen iterativen Prozess aktiv durch
- arbeiten sich selbstständig in fremde Systeme und Frameworks ein
- bearbeiten komplexe Aufgaben ingenieurwissenschaftlich und nehmen eine Aufteilung in Teilaufgaben vor
- organisieren kleinere Projekte und führen diese durch
- geben die Ergebnisse ihrer Arbeit schriftlich und mündlich wieder

##### Sozialkompetenzen

Die Studierenden:

- arbeiten teamorientiert und lösen dabei auch Konflikte
- entwickeln komplexere Software im Team und schätzen deren Aufwände in engerem Rahmen ab (Zeitmanagement)
- reflektieren ihre eigene Leistung und die anderer Studierender (Review und Retrospektive)



### Selbstkompetenzen

Die Studierenden:

- vertiefen ihre Fähigkeit zur Teamarbeit deutlich, insbesondere auch die Fähigkeit zur Konfliktlösung

### Modulinhalte

Im Softwareprojekt entwickelt ein Team von Studierenden über zwei Semester ein größeres Softwaresystem. Dabei wird nach einem iterativen, Scrum-ähnlichen Prozessmodell vorgegangen, wobei typische externe Rollen von den Studierenden innerhalb des Teams wahrgenommen werden. Regelmäßige Präsentationen (pro Semester je 2 vor dem Dozenten, wöchentlich in den Tutorien) sorgen für regelmäßiges Feedback. In einem begleitenden Vorlesungsblock werden die wichtigsten Themen des Software Engineering wiederholt bzw. vertieft und für das Projekt nötige neue Methoden und Techniken bekannt gemacht

### Literaturempfehlungen

#### Links

<https://l.uol.de/swp>

<b>Unterrichtssprache</b>	Deutsch
<b>Dauer in Semestern</b>	2 Semester
<b>Angebotsrhythmus Modul</b>	jährlich, Start immer im Wintersemester
<b>Aufnahmekapazität Modul</b>	unbegrenzt
<b>Lehr-/Lernform</b>	1VL + 1Ü + 1PR
<b>Vorkenntnisse</b>	Programmierung, Datenstrukturen und Algorithmen und Objektorientierte Modellierung und Programmierung

Prüfung	Prüfungszeiten	Prüfungsform
---------	----------------	--------------

#### Gesamtmodul

Die Anmeldung für die Prüfung erfolgt bis zum 1.Dezember im Wintersemester. Die Prüfung erstreckt sich mit unterschiedlichen Teilleistungen bis zum Ende des zweiten Semesters.

Portfolio, bestehend aus:

- Erstellung und Dokumentation von Systemen
- Schriftlicher Kurztest
- Kurzreferat

Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenz
Vorlesung		2	WiSe	28
Übung		2	SoSe und WiSe	28
Projekt		4	SoSe und WiSe	56
<b>Präsenzzeit Modul insgesamt</b>				<b>112 h</b>

---

## inf009 - Praktikum Datenbanken

<b>Modulbezeichnung</b>	Praktikum Datenbanken
<b>Modulkürzel</b>	inf009
<b>Kreditpunkte</b>	6.0 KP
<b>Workload</b>	180 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"><li>• Fach-Bachelor Informatik (Bachelor) &gt; Akzentsetzungsbereich - Wahlbereich Informatik</li><li>• Fach-Bachelor Wirtschaftsinformatik (Bachelor) &gt; Akzentsetzungsbereich Praktische Informatik und Angewandte Informatik</li><li>• Master of Education (Gymnasium) Informatik (Master of Education) &gt; Wahlpflichtmodule (Praktische Informatik)</li><li>• Master of Education (Haupt- und Realschule) Informatik (Master of Education) &gt; Mastermodule</li><li>• Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) &gt; Praktische Vertiefung der Informatik</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Praktische Vertiefung (60 KP)</li></ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"><li>• Grawunder, Marco (Modulverantwortung)</li><li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li></ul>
<b>Teilnahmevoraussetzungen</b>	<p>Nützliche Vorkenntnisse:</p> <ul style="list-style-type: none"><li>• Grundlagen von Datenbanken</li></ul>
<b>Kompetenzziele</b>	<p>Ziele dieses Moduls sind die Vermittlung von praktischen Kenntnissen zu Datenbanken und Informationssystemen. Des Weiteren erlangen die Studierenden einen nachhaltigen Einblick in die technische Realisierung, Implementierung, Installation und Optimierung von Datenbankmanagementsystemen am Beispiel eines professionell eingesetzten DBS.</p> <p><b>Fachkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• verfügen über Kenntnisse zur technischen Realisierung bei der Implementierung und Programmierung von Datenbanksystemen - programmieren und implementieren datenbanknahe Systemroutinen</li><li>• treffen entscheidende Vorgaben in der Modellierungsphase zur Optimierung von Datenbanksystemen - administrieren professionelle Datenbanksysteme (Installation, Verwaltung und Abstimmung)</li><li>• erkennen Performance-Probleme in Datenbanksystemen und beheben diese durch entsprechende Methoden</li><li>• organisieren und steuern Regelabläufe in Datenbanksystemen</li></ul> <p><b>Methodenkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• schlagen für spezielle Anwendungsklassen konkrete Verarbeitungsprinzipien vor</li><li>• reflektieren bestimmte Technologien und Vorgehensweisen bzgl. ihrer Konsequenzen</li></ul> <p><b>Sozialkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• generieren Lösungen zu Problemen von Datenbanksystemen im Team</li></ul> <p><b>Selbstkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• erkennen ihre Belastbarkeit bei der Implementierung und erkennen Fehler</li><li>• reflektieren ihr Selbstbild und ihr Handeln</li></ul>

---

### Modulinhalte

Das Modul Praktikum Datenbanken ist vor allem als praktische Fortführung des Moduls Informationssysteme I konzipiert. Dieses Modul behandelt speziell technische Konzepte eines Datenbanksystems sowie praktische Ansätze in der Datenbankprogrammierung zur Lösung von Optimierungsfragen.

**Schwerpunkte sind dabei:**

- Systemnahes Programmieren auf Datenbankmanagementebene
- Implementierung von Teilaspekten eines Katalogsystems
- Optimierungsstrategien auf Basis unterschiedlicher Anforderungen durch Parallelisierung und Partitionierung von Datenbanken

**Literaturempfehlungen**

- Ramez Elmasri und Shamkant B. Navathe (2007). Fundamentals of Databases Systems. Fifth Edition, Pearson/Addison Wesley
- Held Andrea (2005), Oracle 10g Hochverfügbarkeit Addison-Wesley -
- Held Andrea (2015), Oracle 12c New Features Addison Wesley
- Feuerstein Steven, Pribyl Bill, Dawes Chip (2007).Oracle PL/SQL. 4. Auflage, O'Reillys Taschenbibliothek

**Links**

<b>Unterrichtssprache</b>	Deutsch	
<b>Dauer in Semestern</b>	1 Semester	
<b>Angebotsrhythmus Modul</b>	jedes Wintersemester	
<b>Aufnahmekapazität Modul</b>	unbegrenzt	
<b>Lehr-/Lernform</b>	P	
<b>Vorkenntnisse</b>	Informationssysteme I Betriebssystemkenntnisse	
Prüfung	Prüfungszeiten	Prüfungsform
<b>Gesamtmodul</b>	Über das Semester	Fachpraktische Übung
<b>Lehrveranstaltungsform</b>	Übung	
<b>SWS</b>	4	
<b>Angebotsrhythmus</b>	WiSe	
<b>Workload Präsenzzeit</b>	56 h	

---

## inf014 - Praktikum Betriebssysteme

<b>Modulbezeichnung</b>	Praktikum Betriebssysteme
<b>Modulkürzel</b>	inf014
<b>Kreditpunkte</b>	6.0 KP
<b>Workload</b>	180 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"><li>• Fach-Bachelor Informatik (Bachelor) &gt; Akzentsetzungsbereich - Wahlbereich Informatik</li><li>• Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) &gt; Praktische Vertiefung der Informatik</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Praktische Vertiefung (60 KP)</li></ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"><li>• Theel, Oliver (Modulverantwortung)</li><li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li></ul>
<b>Teilnahmevoraussetzungen</b>	<ul style="list-style-type: none"><li>• Betriebssysteme I</li><li>• Betriebssysteme II</li><li>• Programmiersprachen: C, Assembler</li></ul>
<b>Kompetenzziele</b>	<p>Ziel dieses Moduls ist es praktischen Erfahrungen bei der Analyse, beim Entwurf und bei der Implementierung von relevanten Komponenten eines Betriebssystems sowie deren Zusammenspiel miteinander zu erlernen.</p> <p><b>Fachkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• arbeiten sich in ein komplexes Softwaresystem ein</li><li>• implementieren hardwarenahe Betriebssystem-Komponenten</li><li>• beschreiben die Ausführung paralleler Systemoperationen</li><li>• verstehen die grundlegenden Konzepte der Programmiersprache C++</li><li>• finden systematische Fehler in Software, insbesondere in paralleler Software</li><li>• erarbeiten Aufgabenstellung im Team</li><li>• verwenden UNIX-Standard Programme zum Erstellen von Lösungen</li><li>• erkennen den Vorteil des Arbeitens mit virtuellen Maschinen</li></ul> <p><b>Methodenkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• erkennen Herausforderungen beim Umgang mit Betriebssystemen</li><li>• übertragen Realisierungskonzepte in einen praktischen Kontext</li><li>• hinterfragen unterschiedliche Lösungen kritisch bzgl. ihrer Eigenschaften</li><li>• wählen geeignete Lösungen zur Realisierung aus</li><li>• schreiben ihre Texte mit korrekten deutschen Umlauten</li></ul> <p><b>Sozialkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• lösen die Praktikumsaufgaben in Kleingruppen</li><li>• präsentieren Lösungsvorschläge im Plenum</li><li>• diskutieren ihre unterschiedlichen Lösungsvorschläge innerhalb der Kleingruppen sowie im Plenum</li></ul> <p><b>Selbstkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• nehmen Kritik an</li><li>• organisieren ihre Arbeitsabläufe innerhalb der Kleingruppe</li><li>• reflektieren ihre Lösungsvorschläge unter Berücksichtigung der geäußerten Kritik</li><li>• erkennen eigene Defizite bei der Umsetzung von Theorie in Praxis</li></ul>
<b>Modulinhalte</b>	<p>Das Modul vermittelt folgende Inhalte:</p> <ul style="list-style-type: none"><li>• Analyse eines rudimentären Betriebssystems</li><li>• Konzeption und Implementierung einer Prozessverwaltung</li><li>• Konzeption und Implementierung von Prozesssynchronisationsmechanismen</li></ul>

- Konzeption und Implementierung einer virtuellen Speicherverwaltung
- Konzeption und Implementierung einer Dateiverwaltung oder Benutzerschnittstelle

---

#### Literaturempfehlungen

- Patterson and Hennessy, Computer Organization and Design, 3rd edition, Morgan Kaufmann, 2007

---

#### Links

<b>Unterrichtssprache</b>	Deutsch
<b>Dauer in Semestern</b>	1 Semester
<b>Angebotsrhythmus Modul</b>	jedes Wintersemester
<b>Aufnahmekapazität Modul</b>	unbegrenzt
<b>Hinweise</b>	Verknüpft mit den Modulen: <ul style="list-style-type: none"> <li>• Betriebssysteme I</li> <li>• Betriebssysteme II</li> <li>• Verteilte Systeme</li> </ul>

<b>Lehr-/Lernform</b>	P	
<b>Vorkenntnisse</b>	- Betriebssysteme I - Betriebssysteme II - Programmiersprachen: C, Assembler	
Prüfung	Prüfungszeiten	Prüfungsform
<b>Gesamtmodul</b>	Am Ende des Semesters	Fachpraktische Übungen
<b>Lehrveranstaltungsform</b>	Praktikum	
<b>SWS</b>	4	
<b>Angebotsrhythmus</b>	WiSe	
<b>Workload Präsenzzeit</b>	56 h	

---

## inf018 - Medienverarbeitung

<b>Modulbezeichnung</b>	Medienverarbeitung
<b>Modulkürzel</b>	inf018
<b>Kreditpunkte</b>	6.0 KP
<b>Workload</b>	180 h
<b>Verwendbarkeit des Moduls</b>	

- Fach-Bachelor Betriebswirtschaftslehre mit juristischem Schwerpunkt (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Biologie (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Chemie (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Comparative and European Law (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Engineering Physics (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer" mehr...
- Fach-Bachelor Informatik (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik
- Fach-Bachelor Informatik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Interkulturelle Bildung und Beratung (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Mathematik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Nachhaltigkeitsökonomik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Pädagogik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Pädagogisches Handeln in der Migrationsgesellschaft (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Physik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Physik, Technik und Medizin (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Sozialwissenschaften (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Umweltwissenschaften (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Wirtschaftsinformatik (Bachelor) > Akzentsetzungsbereich Praktische Informatik und Angewandte Informatik
- Fach-Bachelor Wirtschaftsinformatik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Wirtschaftswissenschaften (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Master of Education (Gymnasium) Informatik (Master of Education) > Wahlpflichtmodule (Praktische Informatik)
- Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) > Praktische Vertiefung der Informatik
- Master Wirtschaftsinformatik (Master) > Akzentsetzungsmodulare der Informatik
- Zwei-Fächer-Bachelor Anglistik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Biologie (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Chemie (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Elementarmathematik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Ev. Theologie und Religionspädagogik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Gender Studies (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Germanistik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Geschichte (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Informatik (Bachelor) > PP "Medieninformatik

für Studierende musisch-künstlerischer Fächer"

- Zwei-Fächer-Bachelor Informatik (Bachelor) > Praktische Vertiefung (60 KP)
- Zwei-Fächer-Bachelor Interdisziplinäre Sachbildung (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Kunst und Medien (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Materielle Kultur: Textil (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Mathematik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Musik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Niederdeutsch (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Niederlandistik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Ökonomische Bildung (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Pädagogik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Philosophie / Werte u. Normen (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Physik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Politik-Wirtschaft (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Slavistik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Sonderpädagogik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Sozialwissenschaften (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Sportwissenschaft (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Technik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Wirtschaftswissenschaften (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"

**Zuständige Personen**

- Boll-Westermann, Susanne (Modulverantwortung)
- Lehrenden, Die im Modul (Prüfungsberechtigt)

**Teilnahmevoraussetzungen**

Gute Programmierkenntnisse in Java und/oder C++, Interesse an Medienverarbeitung.

**Kompetenzziele**

Die Studierenden können die Grundlagen der Bildverarbeitung erklären und wissen, welche Algorithmen für die grundlegenden Aufgaben in diesem Bereich existieren und wie diese angewandt werden.  
Die Studierenden sind in der Lage, die in der Vorlesung gelernten grundlegenden Verfahren der Bildverarbeitung zur Lösung einfacher Probleme anzuwenden.

**Fachkompetenzen**

Die Studierenden:

- können die Grundlagen und Eigenschaften von digitalen Medien benennen
- können die zentralen Verfahren zur Kodierung und Kompression von Bildern, Video und Audio erklären
- können grundlegende Verfahren zur Bildverbesserung, Merkmalsextraktion, Merkmalsvermessung, Formanalyse und des Bildverstehens beschreiben

**Methodenkompetenzen**

Die Studierenden:

- können Bildeigenschaften erkennen, beurteilen und über einen sinnvollen Einsatz der Bildverarbeitung entscheiden
- können bestehende Softwarepakete für einfache Probleme der Bildverarbeitung auszuwählen, verwenden und für ihre spezifischen Aufgabenstellung anpassen
- können einfache Bild- und Medienverarbeitungsfunktionen in einer höheren Programmiersprache (z.B. C++) selbständig implementieren

**Sozialkompetenzen**

Die Studierenden:

- können in Kleingruppen ein Softwareprojekt planen, implementieren, und dokumentieren
- können die Ergebnisse ihrer Arbeit anschaulich präsentieren und auf

Kritik und Fragen eingehen

### Selbstkompetenzen

Die Studierenden:

- können während der Entwicklung gemachte Fehler akzeptieren und aus ihnen lernen

<b>Modulinhalte</b>	Das Modul beschäftigt sich mit den Technologien der Medienverarbeitung. Die Vorlesung geht dabei insbesondere auf die Prozesskette von der digitalen Bilderzeugung, über die Bildbearbeitung und Bildspeicherung bis zur Bildanalyse ein. Neben Kompressionsverfahren und Farbraumtheorie (RGB, HSV, YUV, CIEXYZ, ...) umfassen die Themen der Vorlesung dabei Themen der Bildverbesserung, Merkmalsextraktion, Merkmalsvermessung, Formanalyse und des Bildverstehens. Die Vorlesung betrachtet weiterhin die Kodierung und Analyse von Video und Audio.
---------------------	---

<b>Literaturempfehlungen</b>	<ul style="list-style-type: none"><li>• Wilhelm Burger und Mark James Burge. Digitale Bildverarbeitung: Eine Einführung mit Java und Image, J. Springer, 2006.</li><li>• Literatur im Handapparat der Abteilung in der Bibliothek.</li><li>• Linkliste im Lernmanagementsystem zu den einzelnen Themen der Vorlesung.</li></ul>
------------------------------	---

<b>Links</b>	<a href="https://uol.de/medieninformatik/lehrveranstaltungen">https://uol.de/medieninformatik/lehrveranstaltungen</a>
<b>Unterrichtssprache</b>	Deutsch
<b>Dauer in Semestern</b>	1 Semester
<b>Angebotsrhythmus Modul</b>	jedes Wintersemester
<b>Aufnahmekapazität Modul</b>	12
<b>Hinweise</b>	
<b>Lehr-/Lernform</b>	V+Ü
<b>Vorkenntnisse</b>	Gute Programmierkenntnisse in Java und/oder C++, Interesse an Medienverarbeitung.

Prüfung	Prüfungszeiten	Prüfungsform
<b>Gesamtmodul</b>	Die Vorstellung des praktischen Projektes an einem Projekttag aller Kleingruppen findet direkt im Anschluss an die Vorlesungszeit statt. Die mündliche Prüfung findet in den ersten beiden Wochen nach Ende der Vorlesungszeit statt. Etwaige Nachprüfungen finden am Ende der vorlesungsfreien Zeit statt. Der genaue Zeitplan kann den Webseiten der Abteilung sowie den Angaben im Lernmanagementsystem Stud.IP entnommen werden.	Projekt und mündliche Prüfung oder Projekt und Klausur

Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenz
Vorlesung		2	WiSe	28
Übung		2	WiSe	28
<b>Präsenzzeit Modul insgesamt</b>				<b>56 h</b>



---

## inf021 - Fortgeschrittene Java-Technologien

<b>Modulbezeichnung</b>	Fortgeschrittene Java-Technologien
<b>Modulkürzel</b>	inf021
<b>Kreditpunkte</b>	6.0 KP
<b>Workload</b>	180 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"><li>• Fach-Bachelor Informatik (Bachelor) &gt; Akzentsetzungsbereich - Wahlbereich Informatik</li><li>• Fach-Bachelor Wirtschaftsinformatik (Bachelor) &gt; Akzentsetzungsbereich Praktische Informatik und Angewandte Informatik</li><li>• Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) &gt; Praktische Vertiefung der Informatik</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Praktische Vertiefung (60 KP)</li></ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"><li>• Boles, Dietrich (Modulverantwortung)</li><li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li></ul>

### Teilnahmevoraussetzungen

Nützliche Kenntnisse: Objektorientierte Programmierung

### Kompetenzziele

Ziel des Moduls ist es, den Studierenden fortgeschrittene Konzepte und Technologien der Java Plattform Standard Edition (Java SE) zu vermitteln. Die Studierenden sollen die Technologien nach der Veranstaltung selbstständig bei der Entwicklung eigener großer Anwendungen einsetzen können.

#### Fachkompetenzen

Die Studierenden:

- erkennen und benennen die wesentlichen Pakete der JDK-Klassenbibliothek
- strukturieren größere Programme ordentlich und gestalten diese so, dass sie erweiterbar sind
- bauen eigene Klassenbibliotheken auf
- suchen selbstständig in der JDK-Klassenbibliothek nach benötigten Klassen und setzen diese zum Lösen entsprechender Probleme ein
- strukturieren ihre Programme ordentlich
- verstehen und interpretieren auch größere fremde Programme
- beurteilen die Qualität größerer Programme insbesondere in Bezug auf Wartbarkeit, Wiederverwendbarkeit und Erweiterbarkeit

#### Methodenkompetenzen

Die Studierenden:

- recherchieren zur Lösung bestimmter Probleme selbstständig im Internet nach Lösungsansätzen

#### Sozialkompetenzen

Die Studierenden:

- diskutieren mit anderen über eigene und fremde Lösungsansätze

#### Selbstkompetenzen

Die Studierenden:

- reflektieren ihr Vorgehen beim Lösen von Programmierproblemen und nehmen neue Lösungsansätze, z.B. aus dem Internet, in ihr Repertoire auf

### Modulinhalte

Im Rahmen der Vorlesung wird jeweils eine Auswahl folgender Themen vermittelt:

- GUIs (AWT, Swing, JavaFX)
- Java-Basics und Collection-API
- Grafik und Multimedia

- Events
- Model-View-Controller-Prinzip (MVC)
- Threads
- Internationalisierung und Lokalisierung
- Reflection
- IO, Dateien
- Tools (Compiler, Classloader, Drucker, ...)
- Speichertechnologien (XML und Serialisierung)
- Verteilte Programmierung (Sockets, RMI)
- Datenbankenzugriff (JDBC)
- Kompression
- Sicherheitskonzepte

Alternativ wird ein einzelnes Thema ausführlich vertieft.

Im Rahmen der Übungen werden einzelne Programmieraufgaben bzw. durchgängig eine größere Programmieraufgabe bearbeitet. Die Aufgaben haben dabei Bezug zum Thema der einzelnen Vorlesungsinhalte.

---

### Literaturempfehlungen

Linkliste im Lernmanagementsystem

---

### Links

<https://uol.de/medieninformatik/lehveranstaltungen/java-praktikum>

<https://uol.de/medieninformatik/lehveranstaltungen/parallele-programmierung-mit-java-threads>

---

<b>Unterrichtssprache</b>	Deutsch
<b>Dauer in Semestern</b>	1 Semester
<b>Angebotsrhythmus Modul</b>	jedes Semester
<b>Aufnahmekapazität Modul</b>	12
<b>Lehr-/Lernform</b>	V+Ü
<b>Vorkenntnisse</b>	Objektorientierte Programmierung

---

Prüfung	Prüfungszeiten	Prüfungsform
---------	----------------	--------------

---

<b>Gesamtmodul</b>	Durchgängig während des Semesters	Fachpraktische Übungen
--------------------	-----------------------------------	------------------------

---

Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenz
Vorlesung		2	SoSe oder WiSe	56
Übung		2	SoSe oder WiSe	28
<b>Präsenzzeit Modul insgesamt</b>				<b>84 h</b>

---

---

## inf202 - Praktikum Technische Informatik

<b>Modulbezeichnung</b>	Praktikum Technische Informatik
<b>Modulkürzel</b>	inf202
<b>Kreditpunkte</b>	6.0 KP
<b>Workload</b>	180 h
<b>Verwendbarkeit des Moduls</b>	

- Fach-Bachelor Betriebswirtschaftslehre mit juristischem Schwerpunkt (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Biologie (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Chemie (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Comparative and European Law (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Engineering Physics (Bachelor) > Fachnahe Angebote Informatik mehr...
- Fach-Bachelor Informatik (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Interkulturelle Bildung und Beratung (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Mathematik (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Nachhaltigkeitsökonomik (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Pädagogik (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Pädagogisches Handeln in der Migrationsgesellschaft (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Physik (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Physik, Technik und Medizin (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Sozialwissenschaften (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Umweltwissenschaften (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Wirtschaftsinformatik (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Wirtschaftswissenschaften (Bachelor) > Fachnahe Angebote Informatik
- Master of Education (Gymnasium) Informatik (Master of Education) > Wahlpflichtmodule (Technische Informatik)
- Master of Education (Haupt- und Realschule) Informatik (Master of Education) > Mastermodule
- Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) > Praktische Vertiefung der Informatik
- Zwei-Fächer-Bachelor Anglistik (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Biologie (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Chemie (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Elementarmathematik (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Ev. Theologie und Religionspädagogik (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Gender Studies (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Germanistik (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Geschichte (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Informatik (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Informatik (Bachelor) > Praktische Vertiefung (60 KP)
- Zwei-Fächer-Bachelor Interdisziplinäre Sachbildung (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Kunst und Medien (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Materielle Kultur: Textil (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Mathematik (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Musik (Bachelor) > Fachnahe Angebote Informatik

	<ul style="list-style-type: none"> <li>• Zwei-Fächer-Bachelor Niederdeutsch (Bachelor) &gt; Fachnahe Angebote Informatik</li> <li>• Zwei-Fächer-Bachelor Niederlandistik (Bachelor) &gt; Fachnahe Angebote Informatik</li> <li>• Zwei-Fächer-Bachelor Ökonomische Bildung (Bachelor) &gt; Fachnahe Angebote Informatik</li> <li>• Zwei-Fächer-Bachelor Pädagogik (Bachelor) &gt; Fachnahe Angebote Informatik</li> <li>• Zwei-Fächer-Bachelor Philosophie / Werte u. Normen (Bachelor) &gt; Fachnahe Angebote Informatik</li> <li>• Zwei-Fächer-Bachelor Physik (Bachelor) &gt; Fachnahe Angebote Informatik</li> <li>• Zwei-Fächer-Bachelor Politik-Wirtschaft (Bachelor) &gt; Fachnahe Angebote Informatik</li> <li>• Zwei-Fächer-Bachelor Slavistik (Bachelor) &gt; Fachnahe Angebote Informatik</li> <li>• Zwei-Fächer-Bachelor Sonderpädagogik (Bachelor) &gt; Fachnahe Angebote Informatik</li> <li>• Zwei-Fächer-Bachelor Sozialwissenschaften (Bachelor) &gt; Fachnahe Angebote Informatik</li> <li>• Zwei-Fächer-Bachelor Sportwissenschaft (Bachelor) &gt; Fachnahe Angebote Informatik</li> <li>• Zwei-Fächer-Bachelor Technik (Bachelor) &gt; Fachnahe Angebote Informatik</li> <li>• Zwei-Fächer-Bachelor Wirtschaftswissenschaften (Bachelor) &gt; Fachnahe Angebote Informatik</li> </ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"> <li>• Fränzle, Martin Georg (Modulverantwortung)</li> <li>• Janßen, Detlef (Modulverantwortung)</li> <li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li> </ul>
<b>Teilnahmevoraussetzungen</b>	Empfehlung: inf200 „Grundlagen der Technischen Informatik“
<b>Kompetenzziele</b>	<p>Diese Veranstaltung versetzt die Studierenden in die Lage, informationstechnische Systeme zu analysieren, einzelne Komponenten von Rechnern zu verstehen, sie zu entwerfen und zu optimieren sowie qualifiziert über domänenspezifischen Hardwareentwurf zu diskutieren.</p> <p><b>Fachkompetenz</b> Die Studierenden</p> <ul style="list-style-type: none"> <li>• beschreiben einzelne Komponenten von Rechnern</li> <li>• entwerfen und optimieren einzelne Komponenten von Rechnern</li> <li>• entwerfen und optimieren Automaten</li> <li>• spezifizieren und implimentieren autonome Systeme</li> </ul> <p><b>Methodenkompetenz</b> Die Studierenden</p> <ul style="list-style-type: none"> <li>• synthetisieren Rechnerarchitekturen</li> <li>• können Methoden des Hardwareentwurfs auf verschiedene Systeme transferieren</li> </ul> <p><b>Sozialkompetenz</b> Die Studierenden</p> <ul style="list-style-type: none"> <li>• diskutieren qualifiziert über Hardware</li> </ul> <p><b>Selbstkompetenz</b> Die Studierenden</p> <ul style="list-style-type: none"> <li>• sind dazu in der Lage, ihren Kenntnisstand klar gegen Fachkräfte verwandter Disziplinen abzugrenzen</li> </ul>
<b>Modulinhalte</b>	Dieses Modul ist der praktische Teil der Veranstaltung Einführung in die Technische Informatik
<b>Literaturempfehlungen</b>	<ul style="list-style-type: none"> <li>• Skript zur Veranstaltung, Patterson, D.A., Hennesy, J.L.: Computer Organisation and Design: The Hardware/Software Interface</li> </ul>
<b>Links</b>	
<b>Unterrichtssprache</b>	Deutsch
<b>Dauer in Semestern</b>	1 Semester
<b>Angebotsrhythmus Modul</b>	jedes Sommersemester

---

<b>Aufnahmekapazität Modul</b>	unbegrenzt	
<b>Lehr-/Lernform</b>	P	
<b>Vorkenntnisse</b>	Empfehlung: inf200 „Grundlagen der Technischen Informatik	
Prüfung	Prüfungszeiten	Prüfungsform
<b>Gesamtmodul</b>	Am Ende der Vorlesungszeit	Portfolio
<b>Lehrveranstaltungsform</b>	Praktikum	
<b>SWS</b>	4	
<b>Angebotsrhythmus</b>	SoSe	
<b>Workload Präsenzzeit</b>	56 h	

---

---

## inf800 - Proseminar Informatik

<b>Modulbezeichnung</b>	Proseminar Informatik
<b>Modulkürzel</b>	inf800
<b>Kreditpunkte</b>	3.0 KP
<b>Workload</b>	90 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"><li>• Fach-Bachelor Informatik (Bachelor) &gt; Aufbaumodule</li><li>• Fach-Bachelor Wirtschaftsinformatik (Bachelor) &gt; Aufbaucurriculum - Pflichtbereich</li><li>• Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) &gt; Praktische Vertiefung der Informatik</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Praktische Vertiefung (60 KP)</li></ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"><li>• Nieße, Astrid (Modulverantwortung)</li><li>• Sauer, Jürgen (Modulverantwortung)</li><li>• Diethelm, Ira (Modulverantwortung)</li><li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li></ul>
<b>Teilnahmevoraussetzungen</b>	Die spezifischen Teilnahmevoraussetzungen werden in den einzelnen zugeordneten Veranstaltungen beschrieben.
<b>Kompetenzziele</b>	<p>Angeleitet durch einen betreuenden Lehrenden recherchieren Studierende zu einem vorgegebenen Thema nach Literatur, arbeiten sich in diese ein, verstehen und bewerten die Quellen hinsichtlich ihrer Relevanz für das gewählte Thema, präsentieren und diskutieren ihre Erkenntnisse in einem wissenschaftlichen Vortrag und in einer nach wissenschaftlichen Standards aufgebauten Ausarbeitung.</p> <p><b>Fachkompetenzen</b></p> <p>Die Studierenden:</p> <ul style="list-style-type: none"><li>• charakterisieren informatisches Basiswissen (Algorithmenbegriff, Datenstrukturen, Programmierung, Grundlagen der Praktischen, Technischen und Theoretischen Informatik) und wenden dieses an,</li><li>• definieren und beschreiben die wesentlichen</li><li>• mathematischen , logischen und physikalischen Grundlagen der Informatik,</li><li>• definieren und illustrieren differenziert die Kerndisziplinen der Informatik (Theoretische, Praktische und Technische Informatik)</li></ul> <p><b>Methodenkompetenzen</b></p> <p>Die Studierenden:</p> <ul style="list-style-type: none"><li>• begutachten Probleme, formulieren diese mit Hilfe formaler Modelle und untersuchen diese adäquat,</li><li>• untersuchen Probleme anhand technischer und wissenschaftlicher Literatur,</li><li>• reflektieren unter Anleitung ein wissenschaftliches Thema, verfassen angeleitet eine Seminararbeit nach wissenschaftlichen Gesichtspunkten und präsentieren ihre Ergebnisse in einem wissenschaftlichen Vortrag.</li></ul> <p><b>Sozialkompetenzen</b></p> <p>Die Studierenden:</p> <ul style="list-style-type: none"><li>• kommunizieren umsichtig und angemessen mit Anwendern und Fachleuten</li><li>• wenden Präsentationstechniken zielgerichtet an,</li></ul> <p><b>Selbstkompetenzen</b></p> <p>Die Studierenden:</p> <ul style="list-style-type: none"><li>• planen ihr eigenständiges Vorgehen in der Informatik,</li><li>• reflektieren ihre Beiträge kritisch und diskutieren sie mit Anwendern und Fachleuten,</li><li>• ergänzen und vertiefen das im Studium erworbene Wissen selbständig und passen es den aktuellen Entwicklungen des Fachs an.</li></ul>
<b>Modulinhalte</b>	je nach zugeordneter Lehrveranstaltung
<b>Literaturempfehlungen</b>	je nach zugeordneter Lehrveranstaltung

---

**Links**

<b>Unterrichtssprache</b>	Deutsch
<b>Dauer in Semestern</b>	1 Semester
<b>Angebotsrhythmus Modul</b>	jedes Semester
<b>Aufnahmekapazität Modul</b>	unbegrenzt
<b>Hinweise</b>	Aus den angebotenen Seminarveranstaltungen muss eine Veranstaltung belegt werden. Das bestandene Proseminar ist Voraussetzung für die Anmeldung der Bachelorarbeit.

---

<b>Lehr-/Lernform</b>	S
-----------------------	---

---

Prüfung	Prüfungszeiten	Prüfungsform
<b>Gesamtmodul</b>	je nach gewählter Veranstaltung	

- Aktive Mitarbeit im Seminar wird erwartet
- Referat (Ausarbeitung und Vortrag)

---

<b>Lehrveranstaltungsform</b>	Seminar
-------------------------------	---------

---

<b>SWS</b>	2
------------	---

---

<b>Angebotsrhythmus</b>	SoSe oder WiSe
-------------------------	----------------

---

<b>Workload Präsenzzeit</b>	28 h
-----------------------------	------

---

---

## inf803 - Spezielle Themen der Informatik I

<b>Modulbezeichnung</b>	Spezielle Themen der Informatik I
<b>Modulkürzel</b>	inf803
<b>Kreditpunkte</b>	6.0 KP
<b>Workload</b>	180 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"><li>• Fach-Bachelor Informatik (Bachelor) &gt; Akzentsetzungsbereich - Wahlbereich Informatik</li><li>• Fach-Bachelor Wirtschaftsinformatik (Bachelor) &gt; Akzentsetzungsbereich Praktische Informatik und Angewandte Informatik</li><li>• Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) &gt; Praktische Vertiefung der Informatik</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Praktische Vertiefung (60 KP)</li></ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"><li>• Diethelm, Ira (Modulverantwortung)</li><li>• Nieße, Astrid (Modulverantwortung)</li><li>• Sauer, Jürgen (Modulverantwortung)</li><li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li></ul>

### Teilnahmevoraussetzungen

Die erwarteten Vorkenntnisse werden in den Details der zugeordneten Lehrveranstaltung genauer spezifiziert.

### Kompetenzziele

Das Modul hat zum Ziel aktuelle Entwicklungen in der Informatik in den jeweils angemessenen Lehrveranstaltungsformen in das Studium zu integrieren.

#### **Fachkompetenzen**

Die Studierenden:

- kennen neuere technische oder wissenschaftliche Entwicklungen der Informatik
- transferieren Informatik-Methoden und
- Vorgehensmodelle auf die Anforderungen von IT-Anwendungsgebieten
- bewerten die Möglichkeiten und Grenzen informatischer Verfahren und Werkzeuge und setzen diese sachangemessen ein

#### **Methodenkompetenzen**

Die Studierenden:

- begutachten Probleme, formulieren diese mit Hilfe formaler Modelle und untersuchen diese adäquat
- finden (einen oder mehrerer) Lösungszugänge informatischer Probleme und stellen sie dar
- wählen aufgabenangemessene Werkzeuge und Methoden aus und evaluieren diese
- untersuchen Probleme anhand technischer und wissenschaftlicher Literatur

#### **Sozialkompetenzen**

Die Studierenden:

- kooperieren im Team

#### **Selbstkompetenzen**

Die Studierenden:

- planen ihr eigenständiges Vorgehen in der Informatik

### Modulinhalte

Je nach zugeordneten Lehrveranstaltungen

### Literaturempfehlungen

Werden in der zugeordneten Lehrveranstaltung bekannt gegeben.



---

**Links**

<b>Unterrichtsprachen</b>	Deutsch, Englisch
<b>Dauer in Semestern</b>	1 Semester
<b>Angebotsrhythmus Modul</b>	unregelmäßig
<b>Aufnahmekapazität Modul</b>	unbegrenzt

**Hinweise**

Falls mehr dem Modul mehr als eine Veranstaltung zugeordnet ist, wählen Sie in der Regel Veranstaltungen im Gesamtumfang von 4 SWS aus, bspw. eine Vorlesung mit zugehöriger Übung.  
Weitere Informationen finden Sie in der Beschreibung (Details) der zugeordneten Veranstaltungen.

---

<b>Lehr-/Lernform</b>	2 VA aus V, Ü, S, P, PR	
Prüfung	Prüfungszeiten	Prüfungsform
<b>Gesamtmodul</b>		Portfolio oder Referat oder mündliche Prüfung oder Klausur

---

<b>Lehrveranstaltungsform</b>	VA-Auswahl
<b>SWS</b>	4
<b>Angebotsrhythmus</b>	siehe Angebotsrhythmus Modul
<b>Workload Präsenzzeit</b>	56 h

---

---

## inf804 - Spezielle Themen der Informatik II

<b>Modulbezeichnung</b>	Spezielle Themen der Informatik II
<b>Modulkürzel</b>	inf804
<b>Kreditpunkte</b>	6.0 KP
<b>Workload</b>	180 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"><li>• Fach-Bachelor Informatik (Bachelor) &gt; Akzentsetzungsbereich - Wahlbereich Informatik</li><li>• Fach-Bachelor Wirtschaftsinformatik (Bachelor) &gt; Akzentsetzungsbereich Praktische Informatik und Angewandte Informatik</li><li>• Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) &gt; Praktische Vertiefung der Informatik</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Praktische Vertiefung (60 KP)</li></ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"><li>• Diethelm, Ira (Modulverantwortung)</li><li>• Nieße, Astrid (Modulverantwortung)</li><li>• Sauer, Jürgen (Modulverantwortung)</li><li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li></ul>

### Teilnahmevoraussetzungen

Die erwarteten Vorkenntnisse werden in den Details der zugeordneten Lehrveranstaltung genauer spezifiziert.

### Kompetenzziele

Das Modul hat zum Ziel aktuelle Entwicklungen in der Informatik in den jeweils angemessenen Lehrveranstaltungsformen in das Studium zu integrieren.

#### **Fachkompetenzen**

Die Studierenden:

- kennen neuere technische oder wissenschaftliche Entwicklungen der Informatik
- transferieren Informatik-Methoden und -Vorgehensmodelle auf die Anforderungen von IT-Anwendungsgebieten
- bewerten die Möglichkeiten und Grenzen informatischer Verfahren und Werkzeuge und setzen diese sachangemessen ein

#### **Methodenkompetenzen**

Die Studierenden:

- begutachten Probleme, formulieren diese mit Hilfe formaler Modelle und untersuchen diese adäquat
- finden (einen oder mehrerer) Lösungszugänge informatischer Probleme und stellen sie dar
- wählen aufgabenangemessene Werkzeuge und Methoden aus und evaluieren diese
- untersuchen Probleme anhand technischer und wissenschaftlicher Literatur

#### **Sozialkompetenzen**

Die Studierenden:

- kooperieren im Team

#### **Selbstkompetenzen**

Die Studierenden:

- planen ihr eigenständiges Vorgehen in der Informatik

### Modulinhalte

Je nach zugeordneten Lehrveranstaltungen

### Literaturempfehlungen

Werden in der zugeordneten Lehrveranstaltung bekannt gegeben

---

**Links**

<b>Unterrichtsprachen</b>	Deutsch, Englisch
<b>Dauer in Semestern</b>	1 Semester
<b>Angebotsrhythmus Modul</b>	unregelmäßig
<b>Aufnahmekapazität Modul</b>	unbegrenzt

**Hinweise**

Falls mehr dem Modul mehr als eine Veranstaltung zugeordnet ist, wählen Sie in der Regel Veranstaltungen im Gesamtumfang von 4 SWS aus, bspw. eine Vorlesung mit zugehöriger Übung.  
Weitere Informationen finden Sie in der Beschreibung (Details) der zugeordneten Veranstaltungen.

---

<b>Lehr-/Lernform</b>	2 VA aus V, Ü, S, P, PR	
Prüfung	Prüfungszeiten	Prüfungsform
<b>Gesamtmodul</b>		Portfolio oder Referat oder mündliche Prüfung oder Klausur

---

<b>Lehrveranstaltungsform</b>	VA-Auswahl
<b>SWS</b>	4
<b>Angebotsrhythmus</b>	siehe Angebotsrhythmus Modul
<b>Workload Präsenzzeit</b>	56 h

---

---

## inf808 - Aktuelle Themen der Informatik

<b>Modulbezeichnung</b>	Aktuelle Themen der Informatik
<b>Modulkürzel</b>	inf808
<b>Kreditpunkte</b>	3.0 KP
<b>Workload</b>	90 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"><li>• Fach-Bachelor Informatik (Bachelor) &gt; Akzentsetzungsbereich - Wahlbereich Informatik</li><li>• Fach-Bachelor Wirtschaftsinformatik (Bachelor) &gt; Akzentsetzungsbereich Praktische Informatik und Angewandte Informatik</li><li>• Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) &gt; Praktische Vertiefung der Informatik</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Praktische Vertiefung (60 KP)</li></ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"><li>• Diethelm, Ira (Modulverantwortung)</li><li>• Nieße, Astrid (Modulverantwortung)</li><li>• Sauer, Jürgen (Modulverantwortung)</li><li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li></ul>
<b>Teilnahmevoraussetzungen</b>	Keine
<b>Kompetenzziele</b>	<p>Das Modul hat zum Ziel aktuelle Entwicklungen in der Informatik in den jeweils angemessenen Lehrveranstaltungsformen in das Studium zu integrieren.</p> <p><b>Fachkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• kennen neuere technische oder wissenschaftliche Entwicklungen der Informatik</li><li>• transferieren Informatik-Methoden und</li><li>• Vorgehensmodelle auf die Anforderungen von IT-Anwendungsgebieten</li><li>• bewerten die Möglichkeiten und Grenzen informatischer Verfahren und Werkzeuge und setzen diese sachangemessen ein</li></ul> <p><b>Methodenkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• begutachten Probleme, formulieren diese mit Hilfe formaler Modelle und untersuchen diese adäquat</li><li>• finden (einen oder mehrerer) Lösungszugänge informatischer Probleme und stellen sie dar</li><li>• wählen aufgabenangemessene Werkzeuge und Methoden aus und evaluieren diese</li><li>• untersuchen Probleme anhand technischer und wissenschaftlicher Literatur</li><li>• reflektieren unter Anleitung ein wissenschaftliches Thema, verfassen angeleitet eine Seminararbeit nach wissenschaftlichen Gesichtspunkten und präsentieren ihre Ergebnisse in einem wissenschaftlichen Vortrag</li></ul> <p><b>Sozialkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• wenden Präsentationstechniken zielgerichtet an</li></ul> <p><b>Selbstkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• planen ihr eigenständiges Vorgehen in der Informatik</li><li>• reflektieren ihre Beiträge kritisch und diskutieren sie mit Anwendern und Fachleuten</li><li>• ergänzen und vertiefen das im Studium erworbene Wissen selbständig und passen es den aktuellen Entwicklungen des Fachs an</li></ul>
<b>Modulinhalte</b>	Je nach zugeordneten Lehrveranstaltungen

---

**Literaturempfehlungen**

Werden in der zugeordneten Lehrveranstaltung bekannt gegeben

---

**Links**

<b>Unterrichtssprache</b>	Deutsch
<b>Dauer in Semestern</b>	1 Semester
<b>Angebotsrhythmus Modul</b>	unregelmäßig
<b>Aufnahmekapazität Modul</b>	unbegrenzt
<b>Lehr-/Lernform</b>	1 VA aus V, Ü, S, P, PR

Prüfung	Prüfungszeiten	Prüfungsform
---------	----------------	--------------

---

**Gesamtmodul**

Portfolio oder Referat oder mündliche Prüfung oder Klausur

---

<b>Lehrveranstaltungsform</b>	VA-Auswahl
-------------------------------	------------

---

<b>SWS</b>	2
------------	---

---

<b>Angebotsrhythmus</b>	siehe Angebotsrhythmus Modul
-------------------------	------------------------------

---

<b>Workload Präsenzzeit</b>	28 h
-----------------------------	------

---

---

# Wahlpflicht Technische Informatik (30 KP)

## inf200 - Grundlagen der Technischen Informatik

<b>Modulbezeichnung</b>	Grundlagen der Technischen Informatik
<b>Modulkürzel</b>	inf200
<b>Kreditpunkte</b>	6.0 KP
<b>Workload</b>	180 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"><li>• Fach-Bachelor Informatik (Bachelor) &gt; Basismodule</li><li>• Fach-Bachelor Mathematik (Bachelor) &gt; Nebenfachmodule</li><li>• Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) &gt; Akzentsetzungsbereich</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Basismodule</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Wahlpflicht Technische Informatik (30 KP)</li></ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"><li>• Rauh, Andreas (Modulverantwortung)</li><li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li></ul>
<b>Teilnahmevoraussetzungen</b>	Keine Teilnehmvoraussetzungen
<b>Kompetenzziele</b>	<p>Die Studierenden verstehen den Aufbau digitaler Schaltkreise und Rechnersysteme und verfügen über Kenntnisse der grundlegenden technologischen Parameter, Kriterien, Voraussetzungen und Entwicklungen des derzeitigen und zukünftig zu erwartenden Entwurfs digitaler Hardware. Sie verstehen die Grundkonzepte aktueller Rechnerarchitekturen und des Ablaufs von Programmen hierauf. Am Ende der Veranstaltung sollen die Studierenden in der Lage sein, Rechnerarchitekturen zu analysieren, einzelne Hardwarekomponenten von Rechnern zu verstehen, sie zu entwerfen und zu optimieren sowie qualifiziert die Eigenschaften grundlegender Entwurfsalternativen zu diskutieren.</p> <p><b>Fachkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• identifizieren grundlegende Konzepte des Aufbaus digitaler Rechnersysteme, der internen Zahlendarstellung und Schatnetzanalyse sowie deren Optimierung</li></ul> <p><b>Methodenkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• analysieren Rechnerarchitekturen anhand einzelner Komponenten</li><li>• entwerfen und optimieren einzelne Komponenten von Rechnern</li><li>• transferieren systematische Methoden des Schaltkreisentwurfs auf neue Problemstellungen</li></ul> <p><b>Sozialkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• präsentieren ihr Verständnis der Funktionsprinzipien digitaler Rechnersysteme</li></ul> <p><b>Selbstkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• reflektieren die Ergebnisse von Übungsaufgaben kritisch und erkennen Grenzen unterschiedlicher Ansätze für den Entwurf digitaler Rechnersysteme</li></ul>
<b>Modulinhalte</b>	Dieses Modul ist der erste Teil der zweisemestrigen Einführung in die Technische Informatik. Es erläutert die Konstruktionsprinzipien eines Rechners von der Ausführung eines einfachen Programms auf einer Instruction Set Architecture über die grundlegenden Techniken zur Kodierung und Zahlendarstellung, dem Programmablauf auf Masschinenebene, Grundlagen

der Logik und Schaltnetzanalyse sowie deren Optimierung.

### Literaturempfehlungen

- Skript zur Vorlesung
- Schiffmann, W.; Schmitz, R. (2001): Technische Informatik I, II, Übungsbuch; Springer Verlag, Berlin.
- Dal Cin, M. (1996): Rechnerarchitektur; B.G. Teubner.
- Lagemann, K. (1987): Rechnerstrukturen; Springer-Verlag, Berlin.
- Oberschelp, W.; Vossen, G. (1989): Rechneraufbau und Rechnerstrukturen; Oldenbourg-Verlag.
- Mano, Morris M.( 1993): Computer System Architecture 3; Prentice Hall.
- Gajski, D.(1997): Principles of Digital Design; Prentice Hall.
- Patterson, D.A.; Hennessy, J.L. (1997): Computer Organization and Design:
- The Hardware/Software Interface; 2. Edition; Morgan Kaufmann Publishers.
- Wilkinson, B. (1996): Computer Architecture Design and Performance; 2. Edition; Prentice Hall.
- Tanenbaum, A.S.(1999): Structured Computer Organization; 4. Edition; Prentice Hall.

<b>Links</b>					
<b>Unterrichtssprache</b>		Deutsch			
<b>Dauer in Semestern</b>		1 Semester			
<b>Angebotsrhythmus Modul</b>		jährlich			
<b>Aufnahmekapazität Modul</b>		unbegrenzt			
<b>Lehr-/Lernform</b>		1VL + 1Ü			
<b>Vorkenntnisse</b>		keine			
Prüfung	Prüfungszeiten			Prüfungsform	
<b>Gesamtmodul</b>					
		Am Ende der Vorlesungszeit		Klausur oder mündliche Prüfung	
Lehrveranstaltungsform	Kommentar	SWS		Angebotsrhythmus	Workload Präsenz
Vorlesung		3		WiSe	42
Übung		1		WiSe	14
<b>Präsenzzeit Modul insgesamt</b>					56 h

---

## inf201 - Technische Informatik

<b>Modulbezeichnung</b>	Technische Informatik
<b>Modulkürzel</b>	inf201
<b>Kreditpunkte</b>	6.0 KP
<b>Workload</b>	180 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"><li>• Fach-Bachelor Informatik (Bachelor) &gt; Aufbaumodule</li><li>• Master of Education (Gymnasium) Informatik (Master of Education) &gt; Wahlpflichtmodule (Technische Informatik)</li><li>• Master of Education (Haupt- und Realschule) Informatik (Master of Education) &gt; Mastermodule</li><li>• Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) &gt; Akzentsetzungsbereich</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Wahlpflicht Technische Informatik (30 KP)</li></ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"><li>• Rauh, Andreas (Modulverantwortung)</li><li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li></ul>
<b>Teilnahmevoraussetzungen</b>	Kenntnisse des Moduls "Grundlagen der Technischen Informatik"
<b>Kompetenzziele</b>	<p>Die Veranstaltung versetzt die Studierenden in die Lage, Rechnerarchitekturen zu analysieren, einzelne Komponenten von Rechnern zu verstehen, sie zu entwerfen und zu optimieren sowie qualifiziert über domänenspezifischen Hardwareentwurf zu diskutieren.</p> <p><b>Fachkompetenz</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• beschreiben einzelne Komponenten von Rechnern</li><li>• entwerfen und optimieren einzelne Komponenten von Rechnern</li><li>• beschreiben und analysieren elektrische Schaltungen</li></ul> <p><b>Methodenkompetenz</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• analysieren Rechnerarchitekturen</li><li>• verstehen Grundlagen der Analyse and Synthese von Flipflops und Automation</li><li>• verstehen Grundlagen der Analyse elektrischer Schaltungen</li></ul> <p><b>Sozialkompetenz</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• diskutieren qualifiziert über Hardware und Fertigungsprozesse</li><li>• sind in der Lage, ihre Kenntnisse des Hardwareentwurfs auf Domänen außerhalb der Informatik zu übertragen</li></ul> <p><b>Selbstkompetenz</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• reflektieren die Ergebnisse von Übungsaufgaben kritisch und erkennen Grenzen unterschiedlicher Ansätze für den Entwurf digitaler Rechnersysteme</li></ul>
<b>Modulinhalte</b>	Dieses Modul ist der zweite Teil der zweisemestrigen Einführung in die Technische Informatik. Es erläutert sequentielle Schaltungen (z.B. Flipflops und Automaten), arithmetische und logische Rechnerkomponenten, Register und Speicher, Grundlagen der Rechnerkommunikation sowie Grundlagen der Elektrotechnik.
<b>Literaturempfehlungen</b>	<ul style="list-style-type: none"><li>• Skript zur Vorlesung</li><li>• Oberschelp, W., Vossen, G.: Rechneraufbau und Rechnerstrukturen; Oldenbourg Verlag</li><li>• Gajski, D.: Principles of Digital Design; Prentice Hall 1997</li><li>• Patterson, D.A., Hennesy, J.L.: Computer Organisation and Design: The Hardware/Software Interface; 2. Edition; Morgan Kaufman Publishers, 1997</li><li>• Tannenbaum, A.S.: Structured Computer Organization ; 4. Edition; Prentice Hall, 1999</li></ul>
<b>Links</b>	



<b>Unterrichtssprache</b>	Deutsch			
<b>Dauer in Semestern</b>	1 Semester			
<b>Angebotsrhythmus Modul</b>	jährlich			
<b>Aufnahmekapazität Modul</b>	unbegrenzt			
<b>Lehr-/Lernform</b>	V+Ü			
<b>Vorkenntnisse</b>	Kenntnisse des Moduls "Grundlagen der Technischen Informatik"			
<b>Prüfung</b>	<b>Prüfungszeiten</b>	<b>Prüfungsform</b>		
<b>Gesamtmodul</b>	Am Ende der Vorlesungszeit		Klausur oder mündliche Prüfung	
<b>Lehrveranstaltungsform</b>	<b>Kommentar</b>	<b>SWS</b>	<b>Angebotsrhythmus</b>	<b>Workload Präsenz</b>
Vorlesung		3	SoSe	42
Übung		1	SoSe	14
<b>Präsenzzeit Modul insgesamt</b>				<b>56 h</b>

---

## inf203 - Embedded Systems I

<b>Modulbezeichnung</b>	Embedded Systems I
<b>Modulkürzel</b>	inf203
<b>Kreditpunkte</b>	6.0 KP
<b>Workload</b>	180 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"><li>• Fach-Bachelor Informatik (Bachelor) &gt; Akzentsetzungsbereich - Wahlbereich Informatik</li><li>• Master Engineering of Socio-Technical Systems (Master) &gt; Embedded Brain Computer Interaction</li><li>• Master Engineering of Socio-Technical Systems (Master) &gt; Human-Computer Interaction</li><li>• Master Engineering of Socio-Technical Systems (Master) &gt; Systems Engineering</li><li>• Master of Education (Gymnasium) Informatik (Master of Education) &gt; Wahlpflichtmodule (Technische Informatik)</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Wahlpflicht Technische Informatik (30 KP)</li></ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"><li>• Fränzle, Martin Georg (Modulverantwortung)</li><li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li></ul>
<b>Teilnahmevoraussetzungen</b>	<ul style="list-style-type: none"><li>• Grundlagen der technischen Informatik</li><li>• Technische Informatik</li></ul>
<b>Kompetenzziele</b>	<p>Das Modul leistet eine Einführung in den Entwurf digitaler eingebetteter Systeme.</p> <p><b>Fachkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• benennen funktionale und nichtfunktionale Anforderungen zur Spezifikation eingebetteter System</li><li>• diskutieren den Entwurfsraum und der damit verbundenen Entwurfsmethodik eingebetteter Systeme</li><li>• benennen die grundlegenden Verfahren der Steuerungs- und Regelungstechnik</li><li>• charakterisieren die grundlegenden Algorithmen der digitalen Signalverarbeitung</li></ul> <p><b>Methodenkompetenzen</b></p> <ul style="list-style-type: none"><li>• Die Studierenden:</li><li>• konstruieren mit Modellierungswerkzeugen eingebettete Systeme und Regelungssysteme</li><li>• implementieren ein eingebettetes Hardware-/Software-System</li><li>• analysieren verschiedene Spezifikationsprachen anhand diverser Eigenschaften</li></ul> <p><b>Sozialkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• konstruieren Lösungen zu gegebenen Problemen in Gruppen</li><li>• präsentieren Lösungen von informatischen Problemen vor Gruppen</li><li>• organisieren sich zu einer Gruppe zur Lösung eines größeren Problems mit Hilfe gängiger Projektmanagementmethoden</li></ul> <p><b>Selbstkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• erkennen die Grenzen ihrer Belastbarkeit bei der Implementierung von Systemen, bzw. Teilaspekten dieser</li><li>• lösen die Übungsaufgaben eigenverantwortlich</li></ul>
<b>Modulinhalte</b>	<p>Eingebettete Systeme übernehmen komplexe Regelungs-, Steuerungs- oder Datenverarbeitungsaufgaben in technischen Systemen. Sie bilden somit ein wichtiges Wertschöpfungspotential für Produkte der Telekommunikation, der Produktionssteuerung, im Verkehrsbereich und in elektronischen Konsumgütern. Die Funktionalität eingebetteter Systeme wird durch die Integration von Prozessoren, Spezialhardware und Software realisiert. Die Problematik des Entwurfs solcher Systeme ergibt sich durch die Heterogenität</p>

der Systemarchitektur, die Komplexität der Aufgabenstellung und durch die Notwendigkeit, vielfältige technische und ökonomische Vorgaben einhalten zu müssen. In diesem Modul wird zunächst ein Überblick über eingebettete Systeme und den Entwurf dieser Systeme gegeben. Vor allem im Telekommunikations- und Multimedia-Bereich spielt die digitale Signalverarbeitung eine große Rolle. Aus diesem Grund werden danach die Grundlagen und einige Algorithmen der digitalen Signalverarbeitung (z. B. MPEG-Verfahren) vorgestellt. Dagegen sind für verkehrstechnische Anwendungen die Prinzipien der Steuerung und Regelung fundamental, deren Grundlagen ebenfalls dargestellt werden. Anschließend werden die Spezifikation eingebetteter Systeme und die Eigenschaften der dafür eingesetzten Sprachen behandelt. Eine Möglichkeit zur Spezifikation solcher Systeme bieten datenflussorientierte graphische Modellierungssprachen, bspw. Simulink, zusammen mit kontrollflussorientierten graphischen Spezifikationen durch StateCharts, die an dieser Stelle ausführlich vorgestellt werden. Das Modul wird mit einer Behandlung der möglichen Architekturen und der Kommunikationsmodelle in eingebetteten Systemen abgeschlossen. Theoretische sowie praktische Übungen mit dem Entwurfswerkzeug Matlab/Simulink/Stateflow begleiten die Vorlesung und bieten die Möglichkeit den Vorlesungsstoff zu vertiefen.

#### Literaturempfehlungen

#### Foliensammlung sowie:

- Harel, D.: STATECHARTS: A Visual Formalism for Complex Systems. Science of Computer Programming, 8, North-Holland, 1987, page(231-274)
- Harel D.; Naamad, A. The STATEMATE Semantics of Statecharts. ACM Trans. Software Engineering Methods, Oct 1996
- Harel, D.; Politi, M.: Modeling Reactive Systems with Statecharts: The StateMATE Approach
- Josef Hoffmann: Matlab und Simulink: Beispielorientierte Einführung in die Simulation dynamischer Systeme, Addison-Wesley, 1998, ISBN 3-8273-1077-6
- Staunstrup, J., Wolf, W. (eds.): Hardware/Software Co-Design: Principles and Practice. Kluwer Academic Publishers, 1997, ISBN 0-7923-8013-4, chapters 1, 2, (3), 4, 6, (7), (8-10)
- U. Reimers. Digitale Fernsehtechnik. 2. Aufl., Springer, 1997, ISBN 3-540-60945-8

#### Sekundärliteratur:

- Debardeleben, J.A.; Gadiant, A.J.: Incorporating Cost Modeling in Embedded-System Design. IEEE Design & Test, vol 13, no. 3, 1997
- De Michell, G.; Sami, M.: Hardware-Software Co-Design. Kluwer, 1996, ISBN 0-7923-3883-9
- Gajski, D.; Vahid, F.; Narayan, S.; Gong, J.: Specification and Design of Embedded Systems. Prentice Hall, 1994, ISBN 0-13-150731-1
- T. Painter, A. Spanias. Perceptual Coding of Digital Audio. Proceedings of the IEEE, vol 88, no 4, April 2000.
- U. Freyer. DVB Digitales Fernsehen. Verlag Technik, 1997, ISBN 3-341-01192-7
- B. Friedrichs. Kanalcodierung: Grundlagen und Anwendungen in modernen Kommunikationssystemen. Springer, 1995, ISBN 3-540-58232-0
- G.C. Clark. Error-correction coding for digital communications. 3rd printing, Plenum Press, 1988, ISBN 0-306-40615-2
- Artikelserie zum MPEG-2-Standard 3/94 10/94 und das Tutorial "Digitale Bildcodierung" 1/92 1/93, beides in "Fernseh- und Kinotechnik" (BIS: Z elt ZA 1536)

#### Links

Unterrichtssprache	Englisch
Dauer in Semestern	1 Semester
Angebotsrhythmus Modul	jährlich
Aufnahmekapazität Modul	unbegrenzt
Hinweise	Das Modul ist für die Studierenden der Vertiefungsrichtung "Eingebettete Systeme und Mikrorobotik" als Pflichtmodul vorgesehen. <b>Verknüpft mit den Modulen:</b> In dem Modul "Eingebettete Systeme II" werden weitere relevante Themen, wie der Entwurfsprozess, die HW/SW-Partitionierung, die High-Level-Synthese und Hardwarebeschreibungssprachen, diskutiert. Die beiden Module Eingebettete Systeme I und II bieten Querbezüge zu den Modulen "Rechnerarchitektur", "Realzeitbetriebssysteme" und semantikorientierten Modulen der Theoretischen Informatik. Eine Vertiefung der Kenntnisse im Entwurf eingebetteter Systeme ist in den Modulen "System Level Design" und

---

"Low Energy System Design" möglich.

<b>Lehr-/Lernform</b>	V+Ü			
<b>Vorkenntnisse</b>	- Grundlagen der technischen Informatik - Technische Informatik			
<b>Prüfung</b>	<b>Prüfungszeiten</b>	<b>Prüfungsform</b>		
<b>Gesamtmodul</b>	<b>Ende des Semesters</b>	<b>Klausur oder mündliche Prüfung</b>		
<b>Lehrveranstaltungsform</b>	<b>Kommentar</b>	<b>SWS</b>	<b>Angebotsrhythmus</b>	<b>Workload Präsenz</b>
Vorlesung		3	WiSe	42
Übung		1	WiSe	14
<b>Präsenzzeit Modul insgesamt</b>				<b>56 h</b>

---

## inf204 - Embedded Systems II

<b>Modulbezeichnung</b>	Embedded Systems II
<b>Modulkürzel</b>	inf204
<b>Kreditpunkte</b>	6.0 KP
<b>Workload</b>	180 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"><li>• Fach-Bachelor Informatik (Bachelor) &gt; Akzentsetzungsbereich - Wahlbereich Informatik</li><li>• Master Engineering of Socio-Technical Systems (Master) &gt; Embedded Brain Computer Interaction</li><li>• Master Engineering of Socio-Technical Systems (Master) &gt; Systems Engineering</li><li>• Master of Education (Gymnasium) Informatik (Master of Education) &gt; Wahlpflichtmodule (Technische Informatik)</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Wahlpflicht Technische Informatik (30 KP)</li></ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"><li>• Fränzle, Martin Georg (Modulverantwortung)</li><li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li></ul>
<b>Teilnahmevoraussetzungen</b>	Keine Teilnehmvoraussetzungen
<b>Kompetenzziele</b>	<p>Das Modul leistet eine Einführung in den Entwurf digitaler eingebetteter Systeme.</p> <p><b>Fachkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• benennen der Architekturen eingebetteter Systeme</li><li>• benennen spezifische Hardwarekomponenten und -architekturentscheidungen, insbes. Prozessoren</li><li>• charakterisieren den Entwurfsraum und die damit verbundene Entwurfsmethodik eingebetteter Systeme</li><li>• dekomponieren Steuerungs- oder Regelungsaufgaben in Teilkomponenten und setzen diese auf verschiedenen Ebenen des Entwurfsraums um</li><li>• partitionieren und bauen gemischte Software-/Hardwarelösungen auf</li><li>• beschreiben Architekturprinzipien zur Erzielung von Fehlertoleranz</li><li>• beschreiben Analysetechniken zur Bewertung von Echtzeit- und Sicherheitsanforderungen</li><li>• charakterisieren die Formalien der Hardwaresynthese</li></ul> <p><b>Methodenkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• schätzen die Konsequenzen von Entwurfsentscheidungen bzgl. Komponentenallokation und -design in Bezug auf Energieverbrauch, Performanz und Zuverlässigkeit ein</li><li>• implementieren ein eingebettetes Hardware-/Software System anhand einer gegebenen Spezifikation</li><li>• modellieren Hardware mit einer Hardware-Beschreibungs-Sprache</li><li>• analysieren Hardware-/Software Systeme anhand von event basierter Simulation</li></ul> <p><b>Sozialkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• konstruieren Lösungen zu gegebenen Problemen in Gruppen</li><li>• präsentieren Lösungen von Informatischen Problemen vor Gruppen</li><li>• organisieren sich zu einer Gruppe, zur Lösung eines größeren Problems, mithilfe von Projektmanagementmethoden</li></ul> <p><b>Selbstkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• erkennen die Grenzen ihrer Belastbarkeit bei der Implementierung von Systemen, bzw. Teilaspekte dieser</li><li>• beschäftigen sich eigenverantwortlich mit den Übungsaufgaben</li></ul>
<b>Modulinhalte</b>	Eingebettete Systeme übernehmen komplexe Regelungs-, Steuerungs- oder Datenverarbeitungsaufgaben in technischen Systemen. Sie bilden somit ein wichtiges Wertschöpfungspotential für Produkte der Telekommunikation, der Produktionssteuerung, im Verkehrsbereich und in elektronischen Konsumgütern. Die Funktionalität eingebetteter Systeme wird durch die Integration von Prozessoren, Spezialhardware und Software realisiert. Die

Problematik des Entwurfs solcher Systeme ergibt sich durch die Heterogenität der Systemarchitektur, die Komplexität der Aufgabenstellung und durch die Notwendigkeit, vielfältige technische und ökonomische Vorgaben einhalten zu müssen. Dieses Modul baut auf dem Modul Eingebettete Systeme I. In diesem Modul werden zunächst unterschiedliche Architekturen eingebetteter Systeme und der dort eingesetzten Prozessoren behandelt. Gegenstand des Moduls sind anschließend Methoden der Systempartitionierung und der Synthese von Hardware-Komponenten. Theoretische sowie praktische Übungen mit Entwurfswerkzeugen, Hardwarebeschreibungssprachen und Simulationen begleiten die Vorlesung und bieten die Möglichkeit den Vorlesungsstoff zu vertiefen.

#### Literaturempfehlungen

#### Foliensammlung sowie:

- Staunstrup, J.; Wolf, W. (eds.): Hardware/Software Co-Design: Principles and Practice. Kluwer Academic Publishers, 1997, ISBN 0-7923-8013-4, chapters 1, 2, (3), 4, 6, (7), (8-10)
- Yen, Ti-Yen; Wolf, W.: Hardware-Software Co-Synthesis of Distributed Embedded Systems. Kluwer, 1996, ISBN 0-7923-9797-5

#### Gute Sekundärliteratur:

- Peter J. Ashenden: The Designer's Guide to VHDL. Morgan Kaufmann Publishers, 2002, ISBN 1-55860-674-2
- Lehmann, G.; Wunder, B.; Selz, M.: Schaltungsdesign mit VHDL. Franzis Verlag, 1994, ISBN 3-7723-6163-3
- J. Reichardt, B. Schwarz: VHDL-Synthese, Entwurf digitaler Schaltungen und Systeme. Oldenbourg Wissenschaftsverlag, 2000, ISBN 3-486-25128-7
- Mermet, J. (ed.): Fundamentals and Standards in Hardware Description Languages. Kluwer, 1993, ISBN 0-7923-2513-3
- De Micheli, G.; Sami, M.: Hardware-Software Co-Design. Kluwer, 1996, ISBN 0-7923-3883-9
- Gajski, D.; Vahid, F.; Narayan, S.; Gong, J.: Specification and Design of Embedded Systems. Prentice Hall, 1994, ISBN 0-13-150731-1

#### Links

<b>Unterrichtssprache</b>	Englisch			
<b>Dauer in Semestern</b>	1 Semester			
<b>Angebotsrhythmus Modul</b>	jährlich			
<b>Aufnahmekapazität Modul</b>	unbegrenzt			
<b>Hinweise</b>	Das Modul ist für die Studierenden der Vertiefungsrichtung "Eingebettete Systeme und Mikrorobotik" als Pflicht-Modul vorgesehen.			
<b>Lehr-/Lernform</b>	V+Ü			
<b>Vorkenntnisse</b>	keine			
Prüfung	Prüfungszeiten		Prüfungsform	
<b>Gesamtmodul</b>			Klausur oder mündliche Prüfung	
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenz
Vorlesung		3	SoSe	42
Übung		1	SoSe	14
<b>Präsenzzeit Modul insgesamt</b>				56 h

---

## inf205 - Formale Methoden Eingebetteter Systeme

<b>Modulbezeichnung</b>	Formale Methoden Eingebetteter Systeme
<b>Modulkürzel</b>	inf205
<b>Kreditpunkte</b>	6.0 KP
<b>Workload</b>	180 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"><li>• Fach-Bachelor Informatik (Bachelor) &gt; Akzentsetzungsbereich - Wahlbereich Informatik</li><li>• Master of Education (Gymnasium) Informatik (Master of Education) &gt; Wahlpflichtmodule (Technische Informatik)</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Wahlpflicht Technische Informatik (30 KP)</li></ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"><li>• Fränze, Martin Georg (Modulverantwortung)</li><li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li></ul>
<b>Teilnahmevoraussetzungen</b>	Keine
<b>Kompetenzziele</b>	<p>Das Modul vermittelt einen Überblick über semantische Modelle für reaktive Systeme, Echtzeitsysteme und hybride Systeme, sowie Beispiele für entsprechende Spezifikationslogiken. Es erläutert zustandexplorative Verifikationsverfahren sowohl expliziter wie symbolischer Form. Die erworbenen Kenntnisse können überall eingesetzt werden, wo es um die Entwicklung zuverlässiger Software- und Hardwaresysteme geht.</p> <p><b>Fachkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• beurteilen die Tragweite der mit formalen Methoden gewinnbaren Zertifikate fundiert</li><li>• beurteilen die Eignung verfügbarer Verifikationswerkzeuge für eine partikuläre Fragestellung und Systemklasse</li><li>• verwenden diese Werkzeuge an realen Systemen, interpretieren die erzielten Ergebnisse und verbessern in der Folge das untersuchte System zielgerichtet</li><li>• bereiten Systemmodelle für automatische Analyseverfahren vor und abstrahieren bzw. kodieren diese symbolisch (oder anderweitig) entsprechend</li><li>• konzipieren und implementieren eigene Verifikationsalgorithmen</li></ul> <p><b>Methodenkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• beherrschen die mathematische Modellierung komplexer und heterogener Systeme</li><li>• kennen einschlägige mathematische Modelle dynamischer Systeme und können diese auf neue Problemklassen instanzieren</li></ul> <p><b>Sozialkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• entwickeln und implementieren im Team grundlegende Algorithmen der automatischen Verifikation</li><li>• diskutieren die Vor- und Nachteile algorithmischer Alternativen und unterschiedlicher Formalisierungen</li></ul> <p><b>Selbstkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• erkennen ihr fachliches und methodisches Verständnis</li><li>• reflektieren ihre Lösungskompetenz unter Einbezug der vorgestellten Verfahren und Methoden</li></ul>

---

### Modulinhalte

Eingebettete Computersysteme stehen in ständiger Interaktion mit ihrer Umgebung, was zu schwer vorhersehbaren Interaktionssequenzen führen kann. Dieser Umstand erschwert Konstruktion und Validation derartiger Systeme. Vergleichbar dem Einsatz statischer und materialkundlicher Modelle in der Bauwirtschaft sind deshalb formale Modelle für verschiedene Aspekte - z.B. Ausführungszeit, Energiebedarf, mögliche Systemdynamik - eingebetteter Systeme entwickelt worden. Diese stellen den jeweiligen Aspekt des Systems in geschlossener Form dar und erlauben damit die - oft vollautomatische - Herleitung von verlässlichen Kenndaten und Zertifikaten, welche für jedes beliebige Interaktionsszenario mit der Umgebung gelten. Dies steht im Gegensatz zu Methoden des Testens oder Profilings, welche nur ausgewählte Szenarien prüfen und somit nur eine begrenzte Überdeckung bieten können.

In diesem Modul werden verschiedene derartige Modelle erklärt und Methoden zur vollautomatischen Analyse - d.h. Herleitung von Kenndaten oder Zertifikaten - oder Synthese - d.h. automatischen Erzeugung korrekter Systementwürfe - aus derartigen Modellen erläutert und in ihrer Anwendung gezeigt.

In den Übungen besteht die Möglichkeit, die entsprechenden Kenntnisse durch Hands-on-Erfahrung mit domänentypischen Modellierungs- und Verifikationswerkzeugen zu vertiefen, sowie in einem geführten Prozess ein (kleines) vollautomatisches Verifikationswerkzeug selbst zu erstellen.

In der Vorlesung werden die semantischen, logischen und algorithmischen Grundlagen der automatischen Analyse eingebetteter Softwaresysteme vermittelt. Die primäre Unterweisungsform ist hierbei der medial unterstützte Vortrag sowie das didaktische Frage-Antwort-Spiel, wobei als unterstützende Medien Präsentationen, Animationen und Werkzeugvorführungen dienen.

In den Übungen wird das in der Vorlesung erworbene Wissen vertieft und praktisch umgesetzt. Hierzu werden in der ersten Semesterhälfte zweiwöchentlich Übungsaufgaben gestellt, deren Bearbeitung in Kleingruppen zur eigenverantwortlichen Prüfung des Themenverständnisses und zum partnerschaftlichen Lernen anhält.

In der zweiten Semesterhälfte wird eine ebenfalls in Kleingruppen von jeweils 3 Studierenden zu bearbeitende größere Werkzeugentwicklungsaufgabe gestellt, deren Bearbeitung die gesamte Semesterhälfte einnimmt und die Möglichkeit des projektorientierten Lernens bietet. Die Übung dient in dieser Phase der Konsultation mit den Lehrenden; insbesondere werden Lösungsansätze und Probleme vorgestellt und diskutiert.

#### Literaturempfehlungen

- Michael Huth, Mark Ryan: Logic in Computer Science: Modelling and Reasoning About Systems. Cambridge University Press, 2004.
- Christel Baier, Joost-Pieter Katoen: Principles of Model Checking. MIT Press, 2008.
- Edmund M. Clarke, Orna Grumberg, Doron A. Peled: Model Checking. MIT Press, 2000.

#### Links

<b>Unterrichtsprachen</b>	Deutsch, Englisch
<b>Dauer in Semestern</b>	1 Semester
<b>Angebotsrhythmus Modul</b>	jährlich
<b>Aufnahmekapazität Modul</b>	unbegrenzt
<b>Lehr-/Lernform</b>	V+Ü
<b>Vorkenntnisse</b>	Fundierte Grundkenntnisse in mathematischer Logik, diskreter Mathematik, Automaten- und Berechenbarkeitstheorie, wie sie in den Modulen "Diskrete Strukturen" und "Theoretische Informatik I + II" vermittelt werden. Zudem Programmierkenntnisse, wie sie im "Programmierkurs" erworben werden. Begründung: Die in der Vorlesung vorgestellten Verfahren basieren auf einer Operationalisierung von Semantik durch Reduktion auf logische Kodierungen und mechanisierte Prüfung logischer Aussagen. Ein Verständnis dieser Inhalte sowie ihre werkzeugtechnische Umsetzung bedarf der Grundlagen aus den vorgenannten Veranstaltungen.

Prüfung	Prüfungszeiten	Prüfungsform		
<b>Gesamtmodul</b>	1. Termin: Abgabe des Semesterprojekts inkl. schriftlicher Ausarbeitung eine Woche nach Ende der Vorlesungszeit; anschließend Kolloquium und Abschlussgespräch  2. Termin: Wiederholung der Abgabe des Semesterprojekts inkl. schriftlicher Ausarbeitung zwei Wochen vor Beginn des Folgesemesters anschließend Kolloquium und Abschlussgespräch	Projekt Die Prüfungsarbeit besteht aus einem Semesterprojekt sowie dessen Verteidigung in einem Abschlusskolloquium		
<b>Lehrveranstaltungsform</b>	<b>Kommentar</b>	<b>SWS</b>	<b>Angebotsrhythmus</b>	<b>Workload Präsenz</b>
Vorlesung		3	WiSe	42
Übung		1	WiSe	14
<b>Präsenzzeit Modul insgesamt</b>				<b>56 h</b>





---

## inf207 - Grundlagen der Elektrotechnik

<b>Modulbezeichnung</b>	Grundlagen der Elektrotechnik
<b>Modulkürzel</b>	inf207
<b>Kreditpunkte</b>	6.0 KP
<b>Workload</b>	180 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"><li>• Fach-Bachelor Informatik (Bachelor) &gt; Akzentsetzungsbereich - Wahlbereich Informatik</li><li>• Master Informatik (Master) &gt; Interdisziplinäre Module</li><li>• Master of Education (Gymnasium) Informatik (Master of Education) &gt; Wahlpflichtmodule (Technische Informatik)</li><li>• Master of Education (Haupt- und Realschule) Informatik (Master of Education) &gt; Mastermodule</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Wahlpflicht Technische Informatik (30 KP)</li></ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"><li>• Hein, Andreas (Modulverantwortung)</li><li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li></ul>
<b>Teilnahmevoraussetzungen</b>	Modul Analysis II oder Numerik
<b>Kompetenzziele</b>	<p><b>Fachkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• analysieren lineare Netzwerke (Gleich- und Wechselstrom)</li><li>• benennen die Grundkonzepte der Berechnung und Nutzung der Effekte von elektrischen und magnetischen Feldern</li><li>• listen die Eigenschaften einfacher elektrischer Bauelemente (Zweipole) auf</li><li>• berechnen Kenngrößen von einfachen elektrischen Netzwerken</li><li>• setzen computergestützte Analysewerkzeuge ein</li><li>• entwerfen und realisieren einfache Schaltungen</li></ul> <p><b>Methodenkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• übertragen Berechnungsmethoden auf komplexere dynamische Systeme</li><li>• erstellen Modelle elektrischer Systeme</li></ul> <p><b>Sozialkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• präsentieren Lösungen von Problemen</li><li>• vermitteln eigene Lösungen</li></ul> <p><b>Selbstkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• reflektieren ihre Lösungskompetenz unter Einbezug der vorgestellten Verfahren und Methoden</li></ul>
<b>Modulinhalte</b>	<p><b>Das Modul vermittelt die folgenden Inhalte:</b></p> <ul style="list-style-type: none"><li>• Grundbegriffe (elektrische Größen und Einheiten)</li><li>• Netzwerkelemente</li><li>• Berechnung von linearen Gleichstromnetzwerken (Ohmsches Gesetz, Kirchhoffsche Sätze, Überlagerungsprinzip)</li><li>• Größen, Berechnung und Darstellung von elektrischen und magnetischen Feldern</li><li>• Bauelemente (Kondensator und Spule)</li><li>• Erweiterung um zeitabhängige, periodische Größen, Zeigerdarstellung, Berechnungen mit komplexen Effektivwertzeigern</li></ul>
<b>Literaturempfehlungen</b>	<p><b>Essentiell:</b></p> <ul style="list-style-type: none"><li>• Folien zur Vorlesung</li><li>• Albach: Grundlagen der Elektrotechnik 1 und 2. Pearson Studium, 2004.</li></ul> <p><b>Empfohlen:</b></p> <ul style="list-style-type: none"><li>• Hagmann, G.: Grundlagen der Elektrotechnik. AULA-Verlag, 2002. -</li></ul>

- Hagmann, G.: Aufgabensammlung zu den Grundlagen der Elektrotechnik. AULA-Verlag, 2002.

<b>Links</b>				
<b>Unterrichtssprache</b>		Deutsch		
<b>Dauer in Semestern</b>		1 Semester		
<b>Angebotsrhythmus Modul</b>		jährlich		
<b>Aufnahmekapazität Modul</b>		unbegrenzt		
<b>Lehr-/Lernform</b>		V+Ü		
<b>Vorkenntnisse</b>		Modul Analysis II oder Numerik		
Prüfung		Prüfungszeiten	Prüfungsform	
<b>Gesamtmodul</b>		Am Ende des Semesters	Semesterbegleitende fachpraktische Übung und Klausur oder mündliche Prüfung	
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenz
Vorlesung		3	SoSe	42
Übung		1	SoSe	14
<b>Präsenzzeit Modul insgesamt</b>				56 h

---

## inf208 - Mikrorobotik und Mikrosystemtechnik

<b>Modulbezeichnung</b>	Mikrorobotik und Mikrosystemtechnik
<b>Modulkürzel</b>	inf208
<b>Kreditpunkte</b>	6.0 KP
<b>Workload</b>	180 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"><li>• Fach-Bachelor Informatik (Bachelor) &gt; Akzentsetzungsbereich - Wahlbereich Informatik</li><li>• Master Informatik (Master) &gt; Interdisziplinäre Module</li><li>• Master of Education (Gymnasium) Informatik (Master of Education) &gt; Wahlpflichtmodule (Technische Informatik)</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Wahlpflicht Technische Informatik (30 KP)</li></ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"><li>• Fatikow, Sergej (Modulverantwortung)</li><li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li></ul>
<b>Teilnahmevoraussetzungen</b>	Keine Teilnehmervoraussetzungen

---

### Kompetenzziele

Die Mikrorobotik und Mikrosystemtechnik (MST) wird als eine Schlüsseltechnologie mit großem Anwendungspotential, vor allem in der Medizin-, Fertigungs-, Kommunikations-, Bio- und Umwelt- sowie Verkehrstechnik, betrachtet. Trotz des wachsenden Interesses findet man kaum eine Lehrveranstaltung, in der alle wichtigen Bestandteile dieser breitgefächerten Forschungsrichtung behandelt worden wären. Um diese Lücke zu schließen, bietet die Abteilung für Mikrorobotik und Regelungstechnik (AMiR) diese Vorlesung an. Sie soll einen Überblick über die Mikrorobotik und MST, ihre Anwendungsgebiete sowie Lösungsansätze bei der Entwicklung verschiedenartiger Mikrosysteme geben. Die Vorlesung wird durch zahlreiche Beispiele und praktische Ergebnisse veranschaulicht.

#### Fachkompetenzen

Die Studierenden:

- benennen die Ideen, Probleme und Aktivitäten der Mikrorobotik und Mikrosystemtechnik
- beschreiben die Anwendungen der Mikrorobotik und MST
- charakterisieren Verfahren der MST
- beschreiben die Mikromechanik auf Silizium-Basis
- benennen die Prinzipien von Mikrosensoren
- beschreiben Beispiele von Mikrosensoren
- diskutieren informationstechnische Aspekte der MST
- klassifizieren Mikroroboter

#### Methodenkompetenz

Die Studierenden:

- erkennen fächerübergreifende Zusammenhänge sowie funktionale Verknüpfungen von naturwissenschaftlichen und technischen Disziplinen.
- erlernen die technische Abstraktion komplexer Zusammenhänge

#### Sozialkompetenz

Die Studierenden:

- lösen Problemstellungen teilweise in der Gruppe.
- präsentieren der Gruppe ihre Lösungsansätze

#### Selbstkompetenz

Die Studierenden:

- reflektieren ihr Wissen über die technischen Informatik
- erlernen die eigenständige Vertiefung ihrer fachlichen Kompetenz

---

### Modulinhalte

Mikrorobotik und MST:

- Ideen, Probleme, Aktivitäten; Anwendungen der Mikrorobotik und MST;
- Verfahren der MST;
- Mikromechanik auf Silizium-Basis;
- LIGA-Verfahren;

Mikroaktoren:

- Prinzipien und Beispiele (elektrostatistische, piezoelektrische, magnetostruktive, elektromagnetische, Formgedächtnis-, thermomechanische, elektrorheologische und andere Aktoren);

Mikrosensoren:

- Prinzipien und Beispiele (Kraft- und Druck-, Positions- und Geschwindigkeits-, Beschleunigungs-, Bio- und chemische, Temperatur- und andere Sensoren);
- informationstechnische Aspekte der MST;
- Entwurf und Simulation in der MST;
- Klassifikation von Mikrorobotern;
- Grobpositionierung von Mikrorobotern;
- Feinpositionierung von Mikrorobotern; Handhabung von

Mikroobjekten:

- Probleme und Lösungen;
- Mikrogreiftechniken;
- Mikromontage;
- mikroroboterbasierte

Prozessautomatisierung;

- Desktop-Roboterzellen im Rasterelektronenmikroskop.

---

**Literaturempfehlungen**

Essentiell:

- Vorlesungsskript in Buchform wird ausgehändigt (völlig ausreichend für die Prüfung)

---

**Links**

<b>Unterrichtssprache</b>	Deutsch
<b>Dauer in Semestern</b>	1 Semester
<b>Angebotsrhythmus Modul</b>	jährlich
<b>Aufnahmekapazität Modul</b>	unbegrenzt
<b>Lehr-/Lernform</b>	V+Ü
<b>Vorkenntnisse</b>	keine

Prüfung	Prüfungszeiten	Prüfungsform
---------	----------------	--------------

<b>Gesamtmodul</b>	Am Ende der Vorlesungszeit	Mündliche Prüfung auf Deutsch
--------------------	----------------------------	-------------------------------

Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenz
Vorlesung		3	WiSe	42
Übung		1	WiSe	14
<b>Präsenzzeit Modul insgesamt</b>				<b>56 h</b>

---

## inf209 - Regelungstechnik

<b>Modulbezeichnung</b>	Regelungstechnik
<b>Modulkürzel</b>	inf209
<b>Kreditpunkte</b>	6.0 KP
<b>Workload</b>	180 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"><li>• Fach-Bachelor Informatik (Bachelor) &gt; Akzentsetzungsbereich - Wahlbereich Informatik</li><li>• Master Informatik (Master) &gt; Interdisziplinäre Module</li><li>• Master of Education (Gymnasium) Informatik (Master of Education) &gt; Wahlpflichtmodule (Technische Informatik)</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Wahlpflicht Technische Informatik (30 KP)</li></ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"><li>• Fatikow, Sergej (Modulverantwortung)</li><li>• Hein, Andreas (Modulverantwortung)</li><li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li></ul>
<b>Teilnahmevoraussetzungen</b>	<ul style="list-style-type: none"><li>• Differenzialgleichungen</li><li>• Analysis II</li><li>• Grundlagen der Elektrotechnik</li></ul>

---

### Kompetenzziele

#### Fachkompetenzen

Die Studierenden:

- beschreiben grundsätzliche Ansätze zur Steuerung und Regelung von technischen Systemen
- diskutieren die Grundkonzepte der Modellierung von Systemen und deren Kopplung mit Reglern
- benennen die Methoden zur Bestimmung von Qualitätsmerkmalen von geregelten Systemen
- modellieren technische Systeme mit Hilfe von Differenzialgleichungen und deren Umsetzung in Übertragungsfunktionen
- entwerfen Reglerstrukturen, überprüfen deren Stabilität und bestimmen optimale Parameter ihrer Regler

#### Methodenkompetenzen

Die Studierenden:

- erkennen technische Herausforderung und lösen diese durch den Transfer von Realisierungen anderer Disziplinen

#### Sozialkompetenzen

Die Studierenden:

- präsentieren Lösungsansätze

#### Selbstkompetenzen

Die Studierenden:

- arbeiten sich in spezifische Fragen der Entwicklung von geregelten Systemen schnell ein

---

### Modulinhalte

Grundbegriffe

Analoge Übertragungsglieder:

- Lineare zeitinvariante (LZI-) Glieder
- Wirkungspläne
- Simulation und Modellbildung
- Testsignalantworten
- Frequenzgang
- Differentialgleichungen und Übertragungsfunktion
- Stabilität
- Regelstreckenarten
- Reglerarten

**Lineare Regelkreise:**

- Führungs- und Störverhalten
- Stabilitätskriterien
- Klassische Methoden der Analyse und Synthese:
- Realisierung
- Computergestützte Regelung MATLAB/Simulink

**Literaturempfehlungen****Essentiell:**

- Foliensammlung zur Vorlesung

**Empfohlen:**

- Unbehauen, H.: Regelungstechnik I, Klassische Verfahren zur Analyse und Synthese linearer kontinuierlicher Regelsysteme
- Lutz, H. und Wendt, W.: Taschenbuch der Regelungstechnik

**Links**

<b>Unterrichtssprache</b>	Deutsch
<b>Dauer in Semestern</b>	1 Semester
<b>Angebotsrhythmus Modul</b>	jährlich
<b>Aufnahmekapazität Modul</b>	unbegrenzt
<b>Lehr-/Lernform</b>	V+Ü
<b>Vorkenntnisse</b>	- Differenzialgleichungen - Analysis II - Grundlagen der Elektrotechnik

Prüfung	Prüfungszeiten	Prüfungsform
---------	----------------	--------------

<b>Gesamtmodul</b>	Am Ende der Vorlesungszeit	Klausur oder mündliche Prüfung
--------------------	----------------------------	--------------------------------

Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenz
Vorlesung		3	WiSe	42
Übung		1	WiSe	14
<b>Präsenzzeit Modul insgesamt</b>				<b>56 h</b>

---

## inf210 - Signal- und Bildverarbeitung

<b>Modulbezeichnung</b>	Signal- und Bildverarbeitung
<b>Modulkürzel</b>	inf210
<b>Kreditpunkte</b>	6.0 KP
<b>Workload</b>	180 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"><li>• Fach-Bachelor Informatik (Bachelor) &gt; Akzentsetzungsbereich - Wahlbereich Informatik</li><li>• Master Informatik (Master) &gt; Interdisziplinäre Module</li><li>• Master of Education (Gymnasium) Informatik (Master of Education) &gt; Wahlpflichtmodule (Technische Informatik)</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Wahlpflicht Technische Informatik (30 KP)</li></ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"><li>• Hein, Andreas (Modulverantwortung)</li><li>• Fränzle, Martin Georg (Modulverantwortung)</li><li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li></ul>
<b>Teilnahmevoraussetzungen</b>	Modul math040 Analysis II b: Differentialrechnung mehrerer Variablen
<b>Kompetenzziele</b>	<p><b>Fachkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• benennen die Ansätze der Signal- und Bildverarbeitung in technischen Systemen</li><li>• benennen die Methoden/Algorithmen zur Aufbereitung, Filterung, Klassifikation, Interpretation und Visualisierung von Signalen und Bildern</li><li>• wählen Algorithmen abhängig von deren Anwendung und Anforderungen aus</li><li>• schätzen Algorithmen hinsichtlich ihrer Wirksamkeit ein</li><li>• entwerfen Einzelalgorithmen und Verarbeitungsketten und bestimmen deren Qualität</li></ul> <p><b>Methodenkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• arbeiten sich in spezifische Fragen der Signal- und Bildverarbeitung schnell ein</li><li>• Lösungsansätze zu präsentieren</li></ul> <p><b>Sozialkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• präsentieren Lösungsansätze</li><li>• erkennen technische Herausforderungen und reagieren durch Kommunikation mit anderen Disziplinen darauf</li></ul> <p><b>Selbstkompetenzen</b> Die Studierenden:</p> <ul style="list-style-type: none"><li>• reflektieren ihre Lösungen und beziehen dabei die gelernten Methoden ein</li></ul>
<b>Modulinhalte</b>	<p><b>Grundbegriffe</b></p> <ul style="list-style-type: none"><li>• Signalverarbeitung</li><li>• Signalräume und signalverarbeitende Systeme</li><li>• Diskrete und kontinuierliche Signale</li><li>• Kennzeichnung von Signalübertragern anhand von Testsignalen</li><li>• Darstellungsbereiche und Transformationen zwischen diesen</li><li>• Zeitdiskrete Systeme und Abtastung</li><li>• Schätzung und Filterung</li><li>• Konstruktion mit Hilfe von MATLAB</li><li>• Bildverarbeitung</li></ul> <p><b>Einführung/Anwendungsbereiche:</b></p> <ul style="list-style-type: none"><li>• Funktionstransformation</li><li>• Bildverbesserung/Filterung</li><li>• Segmentierung</li><li>• 3D-Rekonstruktion und -Visualisierung</li></ul>

---



## Literaturempfehlungen

### Essentiell:

- Foliensammlung zur Vorlesung

### Empfohlen:

- Meyer, M.; Signalverarbeitung: Analoge und digitale Signale, Systeme und Filter
- Grüningen, D. C. v.; Digitale Signalverarbeitung: mit einer Einführung in die kontinuierlichen Signale und Systeme
- Tönnies, K.; Grundlagen der Bildverarbeitung; Pearson Studium 2005
- Lehmann, Th.; Oberschelp, W.; Pelinak, E.; Peppes, R.; Bildverarbeitung in der Medizin; Springer Verlag 1997
- Handels, H.; Medizinische Bildverarbeitung; Teubner Verlag, Stuttgart Leipzig 2000 weiterführende Literatur wird in der Vorlesung bekannt gegeben

## Links

<b>Unterrichtssprache</b>	Deutsch			
<b>Dauer in Semestern</b>	1 Semester			
<b>Angebotsrhythmus Modul</b>	jährlich			
<b>Aufnahmekapazität Modul</b>	unbegrenzt			
<b>Lehr-/Lernform</b>	V+Ü			
<b>Vorkenntnisse</b>	Modul math040 Analysis II b: Differentialrechnung mehrerer Variablen			
Prüfung	Prüfungszeiten	Prüfungsform		
<b>Gesamtmodul</b>	Am Ende des Semesters	Klausur oder mündliche Prüfung		
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenz
Vorlesung		2	WiSe	28
Übung		2	WiSe	28
<b>Präsenzzeit Modul insgesamt</b>				56 h

---

# Wahlpflicht Theoretische Informatik (30 KP)

## inf400 - Theoretische Informatik: Logik

<b>Modulbezeichnung</b>	Theoretische Informatik: Logik
<b>Modulkürzel</b>	inf400
<b>Kreditpunkte</b>	6.0 KP
<b>Workload</b>	180 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"><li>• Fach-Bachelor Informatik (Bachelor) &gt; Basismodule</li><li>• Master of Education (Gymnasium) Informatik (Master of Education) &gt; Wahlpflichtmodule (Theoretische Informatik)</li><li>• Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) &gt; Akzentsetzungsbereich</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Wahlpflicht Theoretische Informatik (30 KP)</li></ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"><li>• Wehrheim, Heike (Modulverantwortung)</li><li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li></ul>

### Teilnahmevoraussetzungen

Nützliche Vorkenntnisse:

Grundlagen der Mengenlehre, Definition von Funktionen und Relationen

### Kompetenzziele

Einführung in die Aussagenlogik, Prädikatenlogik, Logik-Programmierung und Temporale Logik **Fachkompetenzen**  
Die Studierenden:

- haben Kenntnisse über Syntax, Semantik und Anwendung von Aussagenlogik, Prädikatenlogik, Logik-Programmierung und Temporale Logik
- spezifizieren Probleme mit Hilfe von logischen Formeln
- lösen Fragen über aussagelogische Formeln mit Hilfe von Wahrheitstafeln
- ziehen logische Schlüsse der Aussagen- und Prädikatenlogik mit dem Kalkül des natürlichen Schließens
- beantworten Anfragen an Logik-Programme mit Hilfe der SLD-Resolution
- können Model-Checking von Kripke-Strukturen bezüglich CTL-Formeln algorithmisch durchführen

#### Methodenkompetenzen

Die Studierenden:

- erkennen Logik als ein vielseitiges Hilfsmittel in der Informatik

#### Sozialkompetenzen

Die Studierenden:

- arbeiten in kleinen Gruppen an Lösungen von Aufgaben
- präsentieren Lösungen von Aufgaben vor Gruppen

#### Selbstkompetenzen

Die Studierenden:

- erlernen Ausdauer bei der Bearbeitung schwieriger Aufgaben
- erlernen Präzision beim Aufschreiben von Lösungen

### Modulinhalte

Die Vorlesung führt in die Aussagenlogik, Prädikatenlogik, Logik-Programmierung und Temporale Logik ein. Ein gutes Verständnis von Logik ist für die Informatik von zentraler Bedeutung. Dieses wird bereits durch die weite Verbreitung der logischen Formelsprache in der Informatik belegt. Zum Beispiel kommen einfache Boolesche Ausdrücke in jeder Programmiersprache und beim Schaltkreisentwurf vor; Horn-Klauseln werden zur Wissensrepräsentation eingesetzt; Formeln der Prädikatenlogik und Temporalen Logik werden zum Spezifizieren von Eigenschaften von Soft- und

Hardware benutzt. Neuere Anwendungen wie interaktives und automatisches Beweisen sowie Logik-Programmierung und die damit verwandte Programmiersprache PROLOG unterstreichen den Werkzeugcharakter der Logik in der Informatik. In der Vorlesung werden Syntax, Semantik, Verfahren und Kalküle zur Überprüfung der Gültigkeit von Formeln der Aussagenlogik, Prädikatenlogik und Temporalen Logik eingeführt und an Beispielen illustriert. Zentral ist der Begriff der logischen Folgerung.

**Themen:**

- Aussagenlogik: Syntax und Semantik, Wahrheitstafeln, natürliches Schließen
- Prädikatenlogik: Syntax und Semantik, natürliches Schließen
- Logik-Programmierung: deklarative und prozedurale Semantik, Unifikationsalgorithmus von Robinson, SLD-Resolution, PROLOG
- Temporale Logik CTL: Syntax und Semantik mittels Kripke-Strukturen, Algorithmus zum Model-Checking von CTL

**Literaturempfehlungen**

Essentiell: Skript "Logik"

**Empfohlen:** D. van Dalen: Logic and Structure, Fourth Edition. Springer-Verlag, 2004.

**Gute Sekundärliteratur:** U. Schöning. Logik für Informatiker, Spektrum Verlag, 2000.

**Links**

<b>Unterrichtssprache</b>	Deutsch
<b>Dauer in Semestern</b>	1 Semester
<b>Angebotsrhythmus Modul</b>	jährlich
<b>Aufnahmekapazität Modul</b>	unbegrenzt
<b>Lehr-/Lernform</b>	V+Ü
<b>Vorkenntnisse</b>	keine

Prüfung	Prüfungszeiten	Prüfungsform
<b>Gesamtmodul</b>	Am Ende der Vorlesungszeit	Klausur oder mündliche Prüfung

Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenz
Vorlesung		3	SoSe	42
Übung		1	SoSe	14
<b>Präsenzzeit Modul insgesamt</b>				<b>56 h</b>

---

## inf401 - Grundlagen der Theoretischen Informatik

<b>Modulbezeichnung</b>	Grundlagen der Theoretischen Informatik
<b>Modulkürzel</b>	inf401
<b>Kreditpunkte</b>	6.0 KP
<b>Workload</b>	180 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"><li>• Fach-Bachelor Informatik (Bachelor) &gt; Aufbaumodule</li><li>• Fach-Bachelor Mathematik (Bachelor) &gt; Nebenfachmodule</li><li>• Master of Education (Gymnasium) Informatik (Master of Education) &gt; Pflichtmodule</li><li>• Master of Education (Haupt- und Realschule) Informatik (Master of Education) &gt; Mastermodule</li><li>• Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) &gt; Akzentsetzungsbereich</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Wahlpflicht Theoretische Informatik (30 KP)</li></ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"><li>• Wehrheim, Heike (Modulverantwortung)</li><li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li></ul>

### Teilnahmevoraussetzungen

Nützliche Vorkenntnisse:

Mengenlehre, Funktionen, Relation, Aussagen- und Prädikatenlogik

---

### Kompetenzziele

Einführung in die Theorie der Automaten, formalen Sprachen, Berechenbarkeit und Komplexität

#### **Fachkompetenzen**

Die Studierenden:

- kennen verschiedene Sprachklassen (z.B. reguläre und kontextfreie Sprachen)
- kennen dazugehörige Automatenmodelle (z.B. endliche Automaten, Kellerautomaten, Turingmaschinen)
- erstellen Automaten, Turingmaschinen und Grammatiken zu gegebenen Aufgaben
- kennen äquivalente Formalisierungen des Begriffs des Algorithmus
- weisen Funktionen als algorithmisch berechenbar bzw.
- Probleme als algorithmisch entscheidbar nach
- kennen unentscheidbare Probleme
- schätzen die Komplexität von Algorithmen ab
- kennen Probleme, die deterministisch oder nichtdeterministisch in polynomieller Zeit lösbar sind
- wissen von der Relevanz NP-vollständiger Probleme

#### **Methodenkompetenzen**

Die Studierenden:

- lernen die Mächtigkeit von abstrakten Modellen von Berechenbarkeit kennen
- kennen nicht-effizient lösbare Probleme und können diese in praktischen Aufgabenstellungen wiedererkennen

#### **Sozialkompetenzen**

Die Studierenden:

- arbeiten in kleinen Gruppen an Lösungen von Aufgaben
- präsentieren Lösungen von Aufgaben vor Gruppen

#### **Selbstkompetenzen**

Die Studierenden:

- erlernen Ausdauer bei der Bearbeitung schwieriger Aufgaben
- erlernen Präzision beim Aufschreiben von Lösungen

---

### Modulinhalte

Im ersten Teil der Vorlesung werden verschiedene Sprachklassen (reguläre und kontextfreie Sprachen) eingeführt. Für jede Sprachklasse werden die

dazugehörigen Automatenmodelle (endliche Automaten und Kellerautomaten) vorgestellt, die zum Akzeptieren der jeweiligen Sprachen eingesetzt werden können. Diverse Eigenschaften der eingeführten Sprachen und Automaten werden bewiesen. Im zweiten Teil der Vorlesung wird untersucht, welche Funktionen algorithmisch berechenbar bzw. welche Probleme algorithmisch entscheidbar sind. Dazu wird der Begriff des Algorithmus formalisiert. Turingmaschinen und Grammatiken stellen sich als äquivalente Ansätze heraus. Es wird gezeigt, dass es Probleme gibt, die nicht algorithmisch entscheidbar sind. Dazu gehören auch viele Probleme von praktischem Interesse. Im dritten Teil der Vorlesung geht es um die Komplexität von Algorithmen, d.h. wie viel Zeit und Speicherplatz zum Lösen einer Aufgabe benötigt werden. Insbesondere werden Probleme betrachtet, die deterministisch oder nichtdeterministisch in polynomieller Zeit lösbar sind. Diese Problemklassen sind unter den Namen P und NP bekannt.

## Literaturempfehlungen

Essenziell:

- Skript "Grundbegriffe der Theoretischen Informatik", jeweils in aktueller Ausgabe

Empfohlen:

- Schöning: "Theoretische Informatik kurzgefasst", 5. Auflage, Spektrum, 2008

Gute Sekundärliteratur:

- Hopcroft, Motwani, Ullman: "Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie", Pearson, 2002 (ein Klassiker...)

## Links

<b>Unterrichtssprache</b>	Deutsch
<b>Dauer in Semestern</b>	1 Semester
<b>Angebotsrhythmus Modul</b>	jährlich
<b>Aufnahmekapazität Modul</b>	unbegrenzt
<b>Lehr-/Lernform</b>	V+Ü
<b>Vorkenntnisse</b>	keine

Prüfung	Prüfungszeiten	Prüfungsform
<b>Gesamtmodul</b>	Am Ende des Semesters	Klausur oder mündl. Prüfung

Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenz
Vorlesung		3	WiSe	42
Übung		1	WiSe	14
<b>Präsenzzeit Modul insgesamt</b>				<b>56 h</b>

---

## inf407 - Programmverifikation

<b>Modulbezeichnung</b>	Programmverifikation
<b>Modulkürzel</b>	inf407
<b>Kreditpunkte</b>	6.0 KP
<b>Workload</b>	180 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"><li>• Fach-Bachelor Informatik (Bachelor) &gt; Akzentsetzungsbereich - Wahlbereich Informatik</li><li>• Master of Education (Gymnasium) Informatik (Master of Education) &gt; Wahlpflichtmodule (Theoretische Informatik)</li><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Wahlpflicht Theoretische Informatik (30 KP)</li></ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"><li>• Wehrheim, Heike (Modulverantwortung)</li><li>• Olderog, Ernst-Rüdiger (Modulverantwortung)</li><li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li></ul>
<b>Teilnahmevoraussetzungen</b>	

### Nützliche Vorkenntnisse:

Mengenlehre, Relationen und Funktionen, Aussagen- und Prädikatenlogik, Programmierkenntnisse in einer imperativen Sprache

---

### Kompetenzziele

Einführung in Methoden zum Nachweis der Korrektheit von sequentiellen und parallelen Programmen.

#### Fachkompetenzen

Die Studierenden:

- beschreiben operationelle Semantiken von sequentiellen und parallelen Programmen
- kennen die Konzepte der partiellen und totalen Programmkorrektheit
- zeigen die Korrektheit und Vollständigkeit von Beweissystemen
- stellen Ein-Ausgabe-Spezifikationen von Programmen auf
- führen Korrektheitsbeweise für Programme verschiedener Klassen mit Hilfe von Beweisregeln durch
- überprüfen die Interferenz- und Deadlock-Freiheit paralleler Programme
- führen Korrektheitsbeweise für parallele Programme auf schwachen Speichermodellen durch

#### Methodenkompetenzen

Die Studierenden:

- erkennen Korrektheit als einen wichtigen Aspekt von Programmen und Informatik-Systemen
- beherrschen Kalküle als Mittel der Spezifikation und Verifikation

#### Sozialkompetenzen

Die Studierenden:

- arbeiten in kleinen Gruppen an Lösungen von Aufgaben
- präsentieren Lösungen von Aufgaben vor Gruppen

#### Selbstkompetenzen

Die Studierenden:

- erlernen Ausdauer bei der Bearbeitung schwieriger Aufgaben
- erlernen Präzision bei der Spezifikation von Problemen

---

### Modulinhalte

Programmverifikation ist ein systematischer Ansatz, die Fehlerfreiheit von Programmen zu zeigen. Dazu wird bewiesen, dass ein vorgegebenes Programm bestimmte wünschenswerte Verhaltenseigenschaften besitzt. Beispielsweise sollte ein Sortierprogramm nur sortierte Felder als Ergebnis abliefern. Bei sequentiellen Programmen geht es dabei vor allem um partielle Korrektheit, Terminierung und Abwesenheit von Laufzeitfehlern. Bei parallelen Programmen sind zusätzliche Verhaltenseigenschaften wichtig: Interferenz-

Freiheit, Deadlock-Freiheit und faires Ablaufverhalten. In der Vorlesung geht es vornehmlich um die Verifikation paralleler Programme. Dazu werden klassische Methoden der Hoareschen Logik und ihre Erweiterung für parallele Programme, insbesondere auch für schwache Speichermodelle, studiert. Als Vorbereitung werden zunächst sequentielle Programme behandelt.

### Literaturempfehlungen

Essentiell:

- "K.R. Apt, E.-R. Olderog, Programmverifikation, Springer-Verlag, 1994"

**Oder die erweiterte englische Version:**

- "K.R. Apt, F.S. de Boer, E.-R. Olderog, Verification of Sequential and Concurrent Programs, Third Edition, Springer-Verlag, 2008"

### Links

<b>Unterrichtssprache</b>	Englisch	
<b>Dauer in Semestern</b>	1 Semester	
<b>Angebotsrhythmus Modul</b>	jedes 2. Wintersemester	
<b>Aufnahmekapazität Modul</b>	unbegrenzt	
<b>Lehr-/Lernform</b>	1VL + 1Ü	
<b>Prüfung</b>	Prüfungszeiten	Prüfungsform
<b>Gesamtmodul</b>	Am Ende des Semesters	Klausur oder mündliche Prüfung

Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenz
Vorlesung		3	siehe Angebotsrhythmus Modul	42
Übung		1	siehe Angebotsrhythmus Modul	14
<b>Präsenzzeit Modul insgesamt</b>				<b>56 h</b>

---

# Abschlussmodul

## bam - Bachelorarbeitsmodul

<b>Modulbezeichnung</b>	Bachelorarbeitsmodul
<b>Modulkürzel</b>	bam
<b>Kreditpunkte</b>	15.0 KP
<b>Workload</b>	450 h
<b>Verwendbarkeit des Moduls</b>	<ul style="list-style-type: none"><li>• Zwei-Fächer-Bachelor Informatik (Bachelor) &gt; Abschlussmodul</li></ul>
<b>Zuständige Personen</b>	<ul style="list-style-type: none"><li>• Diethelm, Ira (Modulverantwortung)</li><li>• Lehrenden, Die im Modul (Prüfungsberechtigt)</li></ul>
<b>Teilnahmevoraussetzungen</b>	Keine Teilnehmvoraussetzungen

---

### Kompetenzziele

Die Studierenden zeigen, dass sie in der Lage sind, eine vertiefte wissenschaftlich orientierte Bearbeitung eines Themas der Informatik durchzuführen.

#### **Fachkompetenzen**

Die Studierenden:

- bewerten die Möglichkeiten und Grenzen informatischer Verfahren und Werkzeuge und setzen diese sachgemessen ein.

#### **Methodenkompetenzen**

Die Studierenden:

- wählen aufgabenangemessene Werkzeuge und Methoden aus und evaluieren diese,
- untersuchen Probleme anhand technischer und wissenschaftlicher Literatur,
- führen Softwareprojekte und den Entwurf von Hardware unter Verwendung aktueller Werkzeuge der Informatik durch,
- reflektieren unter Anleitung ein wissenschaftliches Thema, verfassen angeleitet einen Artikel (Seminar- oder Abschlussarbeit) nach wissenschaftlichen Gesichtspunkten und präsentieren ihre Ergebnisse in einem wissenschaftlichen Vortrag.

#### **Sozialkompetenzen**

Die Studierenden:

- erkennen Konflikte und lösen diese im Team,
- wenden Präsentationstechniken und Projektmanagementmethoden zielgerichtet an,
- identifizieren und übernehmen Verantwortung für Aufgaben,
- schätzen die gesellschaftlichen Auswirkungen ihres informatischen Handelns sowie der Informationstechnologie im Allgemeinen ab und hinterfragen diese kritisch,

#### **Selbstkompetenzen**

Die Studierenden:

- wählen sachgemessene, auch eigene Prioritäten aus,
- planen ihr eigenständiges Vorgehen in der Informatik,
- ergänzen und vertiefen das im Studium erworbene Wissen selbständig und passen es den aktuellen Entwicklungen des Fachs an,
- reflektieren ihre Beiträge kritisch und diskutieren sie mit Anwendern und Fachleuten.

---

### Modulinhalte

Ein aktuelles Thema der Informatik wird mit theoretischen, wissenschaftlichen und praktischen Anteilen bearbeitet. Die Ergebnisse werden im Rahmen eines Seminars präsentiert.



---

## Literaturempfehlungen

Nach Vorgabe, themenbezogen.

---

## Links

Bitte in der Studienberatung informieren! Verschiedene Abteilungen haben auch Infos und Formulare auf ihrer Webseite

---

<b>Unterrichtssprache</b>	Deutsch	
<b>Dauer in Semestern</b>	1 Semester	
<b>Angebotsrhythmus Modul</b>	jedes Semester	
<b>Aufnahmekapazität Modul</b>	unbegrenzt	
<b>Lehr-/Lernform</b>	1S	
<b>Prüfung</b>	<b>Prüfungszeiten</b>	<b>Prüfungsform</b>
<b>Gesamtmodul</b>	individuell	Abschlussarbeit, Seminarvortrag
<b>Lehrveranstaltungsform</b>	Seminar	
<b>SWS</b>	2	
<b>Angebotsrhythmus</b>	SoSe und WiSe	
<b>Workload Präsenzzeit</b>	28 h	

---

