



## **Modulhandbuch**

# **Bachelor Studiengang Informatik (B.Sc.)**

Hochschule Landshut  
gültig ab dem Sommersemester 2024

beschlossen am 6. Februar 2024

### **Hinweis:**

Die auf Basis der SPO konkret zu erbringende Prüfungsleistung wird bis spätestens zwei Wochen nach Beginn des Semesters durch die Fakultät Informatik im Studien- und Prüfungsplan hochschulöffentlich bekannt gegeben.

# Inhaltsverzeichnis

IB010	Grundlagen der Informatik	3
IB015	Grundlagen der theoretischen Informatik	4
IB020	Digitaltechnik	5
IB031	Mathematik I	6
IB040	Mathematik II	7
IB061	Software Engineering I	8
IB150	Programmieren I	10
IB250	Programmieren II	12
IB315	Programmieren III	13
IB300	Software Engineering II	14
IB320	Datenbanken	15
IB345	Rechnerarchitektur	16
IB330	Algorithmen und Datenstrukturen	17
IB351	Praxisorientiertes Studienprojekt	18
IB361	IT-Sicherheit	19
IB400	Betriebssysteme	20
IB420	Datenkommunikation	22
IB430	Statistik	23
IB440	Präsentation und Kommunikation	24
IB500	Praktische Zeit im Betrieb	26
IB510	Praxisseminar	27
IB530	IT-Recht	28
IB540	Grundlagen modernes Projektmanagement	29
IB605	Numerik	30
IB610	Compiler	31
IB630	Verteilte Systeme	33
IB640	Internettechnologien	34
IB651	Seminar	35
IB700	Prozessrechentchnik	36
IB7xx	Fachbezogenes Wahlpflichtmodul I - V	38
IB720	Bachelor-Arbeit	39

## Grundlagen der Informatik

IB010

<b>Modulverantwortlicher:</b>	<b>Prof. Dr. Dieter Nazareth</b>
<b>Dozent:</b>	Prof. Dr. Dieter Nazareth
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtmodul
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im ersten Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	-
<b>Voraussetzungen:</b>	-
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	60 Stunden Präsenzzeit im Unterricht 90 Stunden Selbststudium
<b>Lehrformen:</b>	4 SWS seminaristischer Unterricht
<b>Leistungsnachweise und Prüfung:</b>	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben. <b>Die Prüfung zu diesem Modul ist Bestandteil der Grundlagen- und Orientierungsprüfung und muss spätestens am Ende des zweiten Studiensemesters angetreten werden.</b>
<b>Hinweise für dual Studierende:</b>	-

**Qualifikationsziele und Inhalte:****Qualifikationsziele:**

Die Studierenden haben einen Überblick über wichtige Gebiete der Informatik und vertiefte Kenntnisse in ausgewählten Bereichen. Sie kennen die Prinzipien verschiedenartiger Programmiersprachen und Datenstrukturen und sind in der Lage, darauf basierend einfache Algorithmen zu erstellen. Ferner können Sie diese Algorithmen hinsichtlich ihrer Effizienz bewerten.

**Lehrinhalte:**

- Informationssysteme
- Kodierung
- Informelle Algorithmen
- Textersetzung
- Struktogramme
- Funktionale Programmiersprachen
- Prozedurale Programmiersprachen
- Statische Datentypen
- Dynamische Datentypen
- Referenzen
- Komplexität und Berechenbarkeit

**Literatur:**

M. Broy: Informatik 1: Programmierung und Rechnerstrukturen. Springer-Verlag, Berlin 1997.  
H.-P. Grumm, M. Sommer: Einführung in die Informatik, Oldenbourg Verlag, 2004.

## Grundlagen der theoretischen Informatik

IB015

<b>Modulverantwortlicher:</b>	<b>Prof. Andreas Siebert, Ph.D.</b>
<b>Dozent:</b>	Prof. Andreas Siebert, Ph.D., Prof. Dr. Christopher Auer
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im zweiten Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	Kenntnisse aus IB010
<b>Voraussetzungen:</b>	-
<b>Leistungspunkte:</b>	7
<b>Arbeitsaufwand:</b>	60 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit im Praktikum 180 Stunden Selbststudium
<b>Lehrformen:</b>	4 SWS seminaristischer Unterricht 2 SWS Praktikum in kleinen Gruppen (14tägig, 4SWS)
<b>Leistungsnachweise und Prüfung:</b>	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.
<b>Hinweise für dual Studierende:</b>	-

**Qualifikationsziele und Inhalte:****Qualifikationsziele:**

Die Studierenden erkennen den inneren Zusammenhang der grundlegenden Konzepte der Informatik. Sie verstehen die Strukturen und die hierarchischen Beziehungen formaler Sprachen und ihrer äquivalenten Maschinenmodelle sowie deren Auswirkungen auf die Komplexität der zugehörigen Algorithmen. Sie überreißen die Wichtigkeit von probabilistischen Verfahren.

**Lehrinhalte:**

- Automaten
- Formale Sprachen und Grammatiken
- Chomsky-Hierarchie
- Entscheidungsprobleme
- Informationstheorie, Entropie
- Fehlerkorrigierende Codes
- Zufallszahlen
- Warteschlangen
- Theorie in Wissenschaft und Informatik

**Literatur:**

Socher, Rolf: Theoretische Grundlagen der Informatik, Carl Hanser Verlag, 2007.  
 J. Hopcroft, R. Motwani, J. Ullman: Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie, Pearson Studium, 2011.  
 D. Hoffmann, Theoretische Informatik, 5. Auflage, Carl Hanser Verlag, 2022.

## Digitaltechnik

IB020

<b>Modulverantwortlicher:</b>	<b>M.Sc. Thomas Franzke</b>
<b>Dozent:</b>	Prof. Dr. Jürgen Welter
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im ersten Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	-
<b>Voraussetzungen:</b>	-
<b>Leistungspunkte:</b>	3
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit im Unterricht 60 Stunden Selbststudium
<b>Lehrformen:</b>	2 SWS seminaristischer Unterricht und Übungen
<b>Leistungsnachweise und Prüfung:</b>	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.
<b>Hinweise für dual Studierende:</b>	-

**Qualifikationsziele und Inhalte:****Qualifikationsziele:**

Die Studierenden kennen wichtige Schaltnetze und Schaltwerke, die als Grundbausteine in Mikroprozessoren verwendet werden. Sie haben die Fähigkeit einfache Schaltwerke zu entwerfen.

**Lehrinhalte:**

- Schaltalgebra und Schaltsymbole
- Schaltnetze (Kombinatorische Logik):
- Codeumsetzer, Decoder, Demultiplexer, Multiplexer
- Arithmetische Schaltnetze:
- Addierer, Subtrahierer, Arithmetisch-logische Einheit (ALU), Multiplizierer
- Flip-Flops:
- Ungetaktete (asynchrone) Flip-Flops, Getaktete (synchrone) Flip-Flops
- Schaltwerke (Sequentielle Logik):
- Zähler, Register, Schieberegister
- Busse
- Halbleiterspeicher
- Grundstruktur eines Mikroprozessors

**Literatur:**

S. Tanenbaum: „Structured Computer Organization“, Prentice Hall, 2005.  
 Borgmeyer: Grundlagen der Digitaltechnik, Hanser, 2001.  
 Fricke, Digitaltechnik, Vieweg, 2005

## Mathematik I

IB031

<b>Modulverantwortlicher:</b>	<b>Prof. Dr. Michael Sagraloff</b>
<b>Dozent:</b>	Prof. Dr. Sandra Eisenreich, Prof. Dr. Sebastian Schröter, Prof. Dr. Peter Hartmann
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im ersten Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	-
<b>Voraussetzungen:</b>	-
<b>Leistungspunkte:</b>	8
<b>Arbeitsaufwand:</b>	75 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit in den Übungen 135 Stunden Selbststudium
<b>Lehrformen:</b>	5 SWS seminaristischer Unterricht 2 SWS Übungen
<b>Leistungsnachweise und Prüfung:</b>	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.
<b>Hinweise für dual Studierende:</b>	-

**Qualifikationsziele und Inhalte:****Qualifikationsziele:**

Die Studierenden kennen die Gebiete der diskreten Mathematik und der Linearen Algebra, die für das Verständnis der Informatik benötigt werden. Sie haben den Einsatz mathematischer Methoden bei der Lösung von Problemen trainiert. Sie kennen wichtige Anwendungen der oben genannten Gebiete in der Informatik.

**Lehrinhalte:**

Grundbegriffe der Mengenlehre, Aussagen- und Prädikatenlogik, natürliche Zahlen, Induktion und Rekursion, Elemente der Zahlentheorie, Algebraische Strukturen, Kryptographie, Lineare Algebra: Vektorräume, lineare Abbildungen, Matrizen, Gauß'scher Algorithmus, Eigenwerte.

**Literatur:**

Hartmann, Peter: Mathematik für Informatiker, Vieweg-Teubner 2015.  
Teschl, Teschl: Mathematik für Informatiker, Band 1 und 2, Springer 2005.

## Mathematik II

IB040

<b>Modulverantwortlicher:</b>	<b>Prof. Dr. Michael Sagraloff</b>
<b>Dozent:</b>	Prof. Dr. Michael Sagraloff
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im zweiten Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	Mathematik I oder vergleichbare Kenntnisse
<b>Voraussetzungen:</b>	-
<b>Leistungspunkte:</b>	7
<b>Arbeitsaufwand:</b>	60 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit im Praktikum 120 Stunden Selbststudium
<b>Lehrformen:</b>	4 SWS seminaristischer Unterricht und Übungen 2 SWS Praktikum in kleinen Gruppen (14-tägig 4 Stunden)
<b>Leistungsnachweise und Prüfung:</b>	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.
<b>Hinweise für dual Studierende:</b>	-

**Qualifikationsziele und Inhalte:****Qualifikationsziele:**

Die Studierenden haben Kenntnisse der elementaren Funktionen und der Grundlagen der Analysis erworben. Sie haben Einblick in fachbezogene Anwendungen und können mit Mathematiksoftware umgehen.

**Lehrinhalte:**

- Differential- und Integralrechnung mit einer Variablen, elementare Funktionen, Funktionenreihen, Differentialgleichungen, Anwendungen der Mathematik in der Informatik
- Einsatz von Mathematiksoftware zur Lösung mathematischer Probleme. Vertiefung des Lehrstoffes aus der diskreten und der analytischen Mathematik durch das Lösen von Aufgaben am Computer

**Literatur:**

Hartmann, Peter: Mathematik für Informatiker, Vieweg-Teubner 2015.  
Teschl, Teschl: Mathematik für Informatiker, Band 1 und 2, Springer 2005.

## Software Engineering I

IB061

<b>Modulverantwortlicher:</b>	<b>Prof. Dr. Sebastian Schröter</b>
<b>Dozent:</b>	Prof. Dr. Sebastian Schröter
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im zweiten Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	-
<b>Voraussetzungen:</b>	-
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit in den Übungen 90 Stunden Selbststudium
<b>Lehrformen:</b>	2 SWS seminaristischer Unterricht 2 SWS Übungen
<b>Leistungsnachweise und Prüfung:</b>	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.
<b>Hinweise für dual Studierende:</b>	-

**Qualifikationsziele und Inhalte:****Qualifikationsziele:**

Die Studierenden werden für das Thema Software Engineering motiviert. Sie erlangen ein Verständnis für Softwarequalität und erhalten einen Überblick über alle Phasen der Softwareentwicklung. Die Studierenden kennen verschiedene Vorgehensmodelle bei der Softwareentwicklung und erlangen vertiefte Kenntnisse für den Softwareentwurf. Außerdem kennen die Studierenden Testen im Softwarelebenszyklus, Testmetriken, Testmanagement und Testautomatisierung durch Testwerkzeuge wie z.B. JUnit. In den Übungen werden gemeinsam konkrete Fragestellungen beantwortet und ausgesuchte Beispiele bearbeitet.

**Lehrinhalte:**

Motivation und Definition der Begriffe Softwaretechnik, Software Engineering, Softwarequalität usw., Planung (Projektplanung, Aufwandsschätzung, Machbarkeitsstudie, Lastenheft), Anforderungsanalyse (Modellierung, Pflichtenheft), Entwurf (Datenmodellierung, Zustandsmodellierung, Testmetriken, Testautomatisierung, Entity-Relationship Diagramme), Entscheidungstabellen, Softwarearchitektur, Programmierrichtlinien, elementare Grundlagen der analytischen Qualitätssicherung.



**Literatur:**

- Helmut Balzert: Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering. Spektrum Akademischer Verlag, 2009.
- Helmut Balzert: Lehrbuch der Software-Technik: Software-Management. Spektrum Akademischer Verlag, 2008.
- Helmut Balzert: Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb. Spektrum Akademischer Verlag, 2011.
- Ian Sommerville: Software Engineering, 6. Auflage, Verlag Pearson Studium, 2001
- Wolfgang Zuser et al.: Software Engineering mit UML und dem Unified Process, Verlag Pearson Studium, 2001
- Grady Booch et al.: Das UML-Benutzerhandbuch, Addison-Wesley, 1999
- Grady Booch: Objektorientierte Analyse und Design, 2. Auflage, Addison-Wesley, 1996
- Bernd Oestereich: Objektorientierte Softwareentwicklung, Oldenbourg Verlag, 5. Auflage, 2001 A. Spillner, T. Linz: Basiswissen Softwaretest. dpunkt.verlag, 2012
- Peter Liggesmeyer: Software-Qualität: Testen, Analysieren und Verifizieren von Software, Spektrum Akademischer Verlag, 2009.

# Programmieren I

# IB150

<b>Modulverantwortlicher:</b>	<b>Prof. Dr. Dieter Nazareth</b>
<b>Dozent:</b>	Prof. Dr. Dieter Nazareth
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im ersten Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	-
<b>Voraussetzungen:</b>	-
<b>Leistungspunkte:</b>	7
<b>Arbeitsaufwand:</b>	60 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit Praktikum 120 Stunden Selbststudium
<b>Lehrformen:</b>	4 SWS seminaristischer Unterricht 2 SWS Praktikum
<b>Leistungsnachweise und Prüfung:</b>	Leistungsnachweis im Praktikum, Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben. <b>Die Prüfung zu diesem Modul ist Bestandteil der Grundlagen- und Orientierungsprüfung und muss spätestens am Ende des zweiten Studiensemesters angetreten werden.</b>
<b>Hinweise für dual Studierende:</b>	-

## Qualifikationsziele und Inhalte:

### Qualifikationsziele:

Die Studierenden kennen angemessene Verfahren, Methoden und Algorithmen zur Lösung von Problemstellungen einfachen bis mittleren Umfangs in den Programmiersprachen C und C++. Sie sind in der Lage theoretisch erworbenes Wissen planmäßig und systematisch in lauffähige, effiziente Software umzusetzen, die Lösungen angemessen zu testen, sowie strukturelle Schwachstellen zu erkennen und zu beseitigen. Die Studierenden haben verstanden, dass die entwickelten Lösungen -modular, flexibel und kompakt strukturiert sein müssen.

### Lehrinhalte:

- Elementare Datentypen
- Grundlegende Konzepte der prozeduralen Programmierung
- Grundlegende Konzepte der funktionalen Programmierung
- Präprozessoranweisungen
- Ein- und Ausgabe in C und C++
- Datenstrukturen
- Zeiger, Vektoren und Felder
- Grundlegende Konzepte der objektorientierten Programmierung
- Vererbung, Mehrfachvererbung

**Literatur:**

Brian W. Kernighan, Dennis Ritchie: The C Programming Language, Prentice Hall Software, aktuellste Auflage

Jürgen Wolf: C von A bis Z: Das umfassende Handbuch, Galileo Computing, aktuellste Ausgabe

Jürgen Wolf: C++: Das umfassende Handbuch, aktuell zum Standard C++11, Galileo Computing, aktuellste Auflage

# Programmieren II

# IB250

<b>Modulverantwortlicher:</b>	<b>Prof. Dr. Christopher Auer</b>
<b>Dozent:</b>	Prof. Dr. Christopher Auer
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im zweiten Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	Programmieren I
<b>Voraussetzungen:</b>	-
<b>Leistungspunkte:</b>	7
<b>Arbeitsaufwand:</b>	60 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit Praktikum 120 Stunden Selbststudium
<b>Lehrformen:</b>	4 SWS seminaristischer Unterricht 2 SWS Praktikum (jeweils 14-tägig 4 Stunden)
<b>Leistungsnachweise und Prüfung:</b>	Leistungsnachweis im Praktikum, Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.
<b>Hinweise für dual Studierende:</b>	-

## Qualifikationsziele und Inhalte:

### Qualifikationsziele:

Die Studierenden können angemessene Verfahren, Methoden und Algorithmen zur Lösung von Problemstellungen einfachen bis mittleren Umfangs in der Programmiersprache Java anwenden. Sie sind in der Lage theoretisch erworbenes Wissen planmäßig und systematisch in lauffähige, effiziente Software umzusetzen, sowie strukturelle Schwachstellen zu erkennen und zu beseitigen. Die Studierenden erwerben ein Verständnis dafür, wie Softwarelösungen modular, flexibel und kompakt zu gestalten sind.

### Lehrinhalte:

- Java Laufzeitsystem, Garbage Collection
- Java Typsystem
- Grundlegende Konzepte der objektorientierten Programmierung, Vererbung und Beziehungen zwischen Typen.
- Datenkapselung, Immutability, Konzepte von Gleichheit und Identität
- Entwicklung von Lösungen für konkrete Problemstellungen und Umsetzung der Lösungsideen in lauffähige Software unter Einhaltung professioneller Maßstäbe und Kriterien
- Einsatz von Klassenbibliotheken und Umgang mit Fehlern
- Ein- und Ausgabe
- Definition und Nutzung von Container-Datenstrukturen

### Literatur:

Reinhard Schiedermeier: Programmieren mit Java. Pearson 2010.  
 Reinhard Schiedermeier: Programmieren mit Java II. Pearson 2013.  
 Michael Inden, Der Weg zum Java-Profi, dpunkt-Verlag 2015  
 Dan Pilone, Russ Miles: Head First Software Development. O'Reilly 2008  
 Reinhard Schiedermeier, Klaus Köhler: Das Java Praktikum, d-punkt-Verlag 2008

# Programmieren III

# IB315

<b>Modulverantwortlicher:</b>	<b>Prof. Dr. Christopher Auer</b>
<b>Dozent:</b>	Prof. Dr. Christopher Auer
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im dritten Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	Programmieren I und II
<b>Voraussetzungen:</b>	-
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit Praktikum 90 Stunden Selbststudium
<b>Lehrformen:</b>	2 SWS seminaristischer Unterricht 2 SWS Praktikum (jeweils 14-tägig 4 Stunden)
<b>Leistungsnachweise und Prüfung:</b>	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.
<b>Hinweise für dual Studierende:</b>	-

## Qualifikationsziele und Inhalte:

### Qualifikationsziele:

Die Studierenden erwerben die Fähigkeit, theoretisch erfasste Verfahren, Methoden und Algorithmen in lauffähige und effiziente Software umzusetzen, die Lösungen angemessen zu testen, sowie strukturelle Schwachstellen zu erkennen und zu beseitigen. Ferner kennen sie grundlegende Datenstrukturen, können einfache Datenstrukturen selbst implementieren und vorhandene Klassen aus etablierten Klassenbibliotheken und Frameworks zielgerichtet zur Lösung von komplexeren Problemstellungen einsetzen. Sie haben einen ersten Eindruck vom Komplexitätsbegriff und können die Tragweite von Tests abschätzen. Die Studierenden bilden Verständnis dafür aus, dass technisches Funktionieren nicht ausreicht, denn die entwickelten Lösungen müssen modular, flexibel und kompakt strukturiert sein.

### Lehrinhalte:

- Nebenläufigkeit
- Typabstraktion (Generics)
- Funktionale Programmierung und konkrete Anwendungen (Lambdas und Streams)
- Testen, Softwagemetriken, Qualitätskriterien

### Literatur:

Reinhard Schiedermeier: Programmieren mit Java. Pearson 2010.  
 Reinhard Schiedermeier: Programmieren mit Java II. Pearson 2013.  
 James Gosling: The Java Language Specification, Addison-Wesley 2005.  
 Joshua Bloch: Effektiv Java programmieren, Addison-Wesley 2002.  
 Barbara Liskov: Program Development in Java. Addison-Wesley 2002  
 Robert C. Martin: Clean Code. Prentice Hall 2009  
 Michael Inden: Java 9 Die Neuerungen, dpunkt-Verlag 2018

## Software Engineering II

IB300

<b>Modulverantwortlicher:</b>	<b>Prof. Dr. Abdelmajid Khelil</b>
<b>Dozent:</b>	Prof. Dr. Abdelmajid Khelil
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im dritten Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	Software Engineering I, Programmieren I oder vergleichbare Kenntnisse
<b>Voraussetzungen:</b>	Zulassung zum Praktikum erfolgt bei bestandener Prüfung in Programmieren I oder Programmieren II
<b>Leistungspunkte:</b>	7
<b>Arbeitsaufwand:</b>	60 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit Praktikum 120 Stunden Selbststudium
<b>Lehrformen:</b>	4 SWS seminaristischer Unterricht 2 SWS Praktikum in kleinen Gruppen (14tägig 4 Stunden)
<b>Leistungsnachweise und Prüfung:</b>	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.
<b>Hinweise für dual Studierende:</b>	-

**Qualifikationsziele und Inhalte:****Qualifikationsziele:**

Die Studierenden können komplexe, umfangreiche Softwareprojekte systematisch mit ingenieurmäßigen Methoden durchführen. Sie kennen die existierenden und aktuellen Modellierungsmöglichkeiten und die Vorgehensweise bei der Entwicklung von Software. Sie haben Kenntnis über Design Patterns und können sie in Projekten einsetzen.

**Lehrinhalte:**

- Wichtigste Elemente und Diagramme der UML und deren Anwendung in der Softwareentwicklung, Vorgehen bei der objektorientierten Softwareentwicklung und Modellierung unter Einsatz von UML.
- Analysemuster, Design Patterns und deren Einsatz
- Structured Analysis, Realtime Analysis, Structured Design

**Literatur:**

RuppZengler/Queins: UML2 glasklar, 3. Auflage Hanser 2007

## Datenbanken

IB320

<b>Modulverantwortlicher:</b>	<b>M.Sc. Thomas Franzke</b>
<b>Dozent:</b>	Prof. Dr. Reimer Studt
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im dritten Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	Erster Studienabschnitt oder vergleichbare Kenntnisse, Grundkenntnisse in Java.
<b>Voraussetzungen:</b>	
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit Praktikum 90 Stunden Selbststudium
<b>Lehrformen:</b>	2 SWS seminaristischer Unterricht 2 SWS Praktikum in kleinen Gruppen (14tägig 4 Stunden)
<b>Leistungsnachweise und Prüfung:</b>	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.
<b>Hinweise für dual Studierende:</b>	-

**Qualifikationsziele und Inhalte:****Qualifikationsziele:**

Die Studierenden besitzen theoretische und praktische Kenntnisse über relationale, objektrelationale und NoSQL-Datenbanken.

**Lehrinhalte:**

- Aufbau und Funktionen eines Datenbanksystems
- Datenbankentwurf: Entity-Relationship-Modell, Normalisierung
- Relationales Datenbank-Modell
- Anfragesprachen: relationale Algebra, Structured Query Language (SQL)
- Indexstrukturen in relationalen Datenbanken
- Transaktionen, Trigger, Query-Optimierung
- eingebettetes SQL, Java Database Connectivity (JDBC)
- NoSQL-Datenbanken (MongoDB)

**Literatur:**

R. Elmasri, S. B. Navathe: Grundlagen von Datenbanksystemen, Addison-Wesley

## Rechnerarchitektur

IB345

<b>Modulverantwortlicher:</b>	<b>Prof. Dr. Markus Mock</b>
<b>Dozent:</b>	Prof. Dr. Markus Mock
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im vierten Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	-
<b>Voraussetzungen:</b>	Zulassung zum Praktikum erfolgt bei bestandener Prüfung in Programmieren I oder Programmieren II
<b>Leistungspunkte:</b>	7
<b>Arbeitsaufwand:</b>	60 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit im Praktikum 120 Stunden Selbststudium
<b>Lehrformen:</b>	4 SWS seminaristischer Unterricht 2 SWS Praktikum in kleinen Gruppen (14tägig 4 Stunden)
<b>Leistungsnachweise und Prüfung:</b>	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.
<b>Hinweise für dual Studierende:</b>	-

**Qualifikationsziele und Inhalte:****Qualifikationsziele:**

Die Studierenden sind mit den grundlegenden Ideen und Technologien der Rechnerorganisation vertraut. Sie verstehen den Aufbau moderner Prozessoren und dessen Auswirkungen auf die Laufzeiteigenschaften von Programmen. Sie sind in der Lage, diese Einsichten zum Verstehen und Optimieren des Laufzeitverhaltens von Programmen zu nutzen

**Lehrinhalte:**

Architektur und Organisation moderner Prozessoren mit den Schwerpunkten Speicherverwaltung, Cache und Parallelität:

- Befehlssprache von Prozessoren, mit Schwerpunkt MIPS
- Unterstützung von Programmiersprachenkonzepten durch Hardware
- Computerarithmetik
- Speicherverwaltung und Speicherhierarchie, Cache, virtueller Speicher
- Virtuelle Maschinen
- Probleme und Chancen der Parallelverarbeitung, Pipelining und superskalare Prozessoren

**Literatur:**

J. Hennessy, D. Patterson: Rechnerorganisation und – Entwurf, ab 4ter Auflage



## Algorithmen und Datenstrukturen

IB330

<b>Modulverantwortlicher:</b>	<b>Prof. Andreas Siebert, Ph.D.</b>
<b>Dozent:</b>	Prof. Andreas Siebert, Ph.D.
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im vierten Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	Programmierkenntnisse in Java oder C/C++
<b>Voraussetzungen:</b>	-
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit im Praktikum 90 Stunden Selbststudium
<b>Lehrformen:</b>	2 SWS seminaristischer Unterricht 2 SWS Praktikum in kleinen Gruppen (14tägig 4 Stunden)
<b>Leistungsnachweise und Prüfung:</b>	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.
<b>Hinweise für dual Studierende:</b>	-

**Qualifikationsziele und Inhalte:****Qualifikationsziele:**

Die Studierenden kennen die wesentlichen Datenstrukturen und Algorithmen, die für die Softwareentwicklung benötigt werden. Sie haben die Implementierung von effizienten Algorithmen eingeübt. Sie haben ein Verständnis für die asymptotische Laufzeitkomplexität von Algorithmen entwickelt und können sie analytisch herleiten.

**Lehrinhalte:**

- Komplexität von Algorithmen, Landau-Symbole, Master-Theorem
- Paradigmen der Algorithmenentwicklung
- NP-vollständige Probleme
- Randomisierte Algorithmen
- Parallele Algorithmen
- Ausgewählte Algorithmen

**Literatur:**

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein: An Introduction to Algorithms, 4th ed., The MIT Press, 2022.

Peter Sanders, Kurt Mehlhorn, Martin Dietzfelbinger, Roman Dementiev: Sequential and Parallel Algorithms and Data Structures, Springer, 2019.

## Praxisorientiertes Studienprojekt

IB351

<b>Modulverantwortlicher:</b>	<b>M.Sc. Thomas Franzke</b>
<b>Dozent:</b>	Dozenten der Fakultät Informatik
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	Im sechsten Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	Programmieren I, Software Engineering I, Grundlagen der Informatik
<b>Voraussetzungen:</b>	Zulassung erfolgt bei bestandener Prüfung in Programmieren I oder Programmieren II
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	60 Stunden nicht ständig betreute Projektarbeit im Labor 90 Stunden eigenverantwortliches Arbeiten am Projekt
<b>Lehrformen:</b>	4 SWS nicht ständig betreute Projektarbeit im Labor Eigenverantwortliches Arbeiten der Studierenden in Teams von einer kritischen Größe, so dass das Auftreten typischer Schnittstellenprobleme gewährleistet ist.
<b>Leistungsnachweise und Prüfung:</b>	Benoteter Leistungsnachweis durch individuelle schriftliche Ausarbeitung jedes Teammitglieds zum eigenen Beitrag im Projekt, im Team erstellte Gesamtdokumentation.
<b>Hinweise für dual Studierende:</b>	-

**Qualifikationsziele und Inhalte:****Qualifikationsziele:**

Die Studierenden kennen die Problematik der Erstellung komplexer Systeme. Sie können die Grundlagen des wissenschaftlichen Arbeitens anwenden und wissen, wie man eigenverantwortlich dem Studiengang entsprechende Projekte durchführt. Sie haben Teamarbeit trainiert und Kenntnisse in der Abschätzung des Umfangs von Projekten sowie in Management und Kontrolle von Projekten erworben. Sie sind in der Lage, fachübergreifende Kenntnisse anzuwenden und Arbeitsergebnisse zu präsentieren.

**Lehrinhalte:**

Die Lehrenden der Fakultät Informatik bieten den Studierenden per Aushang Projektthemen mit einer kurzen Beschreibung zur Auswahl an. Teams von Studenten können selbst ein Projekt vorschlagen, dafür müssen Sie einen Betreuungsdozenten finden. Die Studenten werden von dem ausgebenden Dozenten regelmäßig fachlich betreut.

Projekte im Rahmen des Moduls "Unternehmerische Kompetenzen" (Campus Company) können ebenfalls anerkannt werden, sofern das Thema des Projekts zum Studiengang passt.

**Literatur:**

Siehe Projektbeschreibungen.

## IT-Sicherheit

IB361

<b>Modulverantwortlicher:</b>	<b>Prof. Dr. Johann Uhrmann</b>
<b>Dozent:</b>	Prof. Dr. Johann Uhrmann
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im dritten Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	Programmieren I
<b>Voraussetzungen:</b>	-
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit in der Übung 90 Stunden Selbststudium
<b>Lehrformen:</b>	2 SWS seminaristischer Unterricht 2 SWS Übung in kleinen Gruppen (14tägig)
<b>Leistungsnachweise und Prüfung:</b>	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.
<b>Hinweise für dual Studierende:</b>	-

**Qualifikationsziele und Inhalte:****Qualifikationsziele:**

Erkennen von Risiken in der Informationsgesellschaft. Kenntnis wichtiger Dienste und Mechanismen zur Erstellung und zum Einsatz sicherer IT-Systeme.

**Lehrinhalte:**

Analyse von Sicherheitsbedrohungen.

Die Säulen der IT-Sicherheit: Verfügbarkeit, Vertraulichkeit, Integrität, Verbindlichkeit. Sicherheitsbedrohungen.

Sicherheitsbasisdienste: Kryptographie, Key Management, Authentifizierung.

Sicherheitsarchitekturen und Protokolle: pgp, S/MIME, TLS. Firewalls.

Cloud-Security.

Aktuelle Entwicklungen in der IT-Sicherheit

**Literatur:**

Roland Hellmann, IT-Sicherheit - Eine Einführung, De Gruyter, 2018.

Michael Messner, Hacking mit Metasploit, dpunkt, 2017.

Claudia Eckert, IT-Sicherheit: Konzepte - Verfahren - Protokolle, De Gruyter, 2023.

Weitere Literatur wird in der Veranstaltung bekanntgegeben.

## Betriebssysteme

IB400

<b>Modulverantwortlicher:</b>	<b>Prof. Dr. Martin Pellkofer</b>
<b>Dozent:</b>	Prof. Dr. Martin Pellkofer
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im vierten Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	Grundlagen der Informatik, Programmieren I und II, SW-Engineering I
<b>Voraussetzungen:</b>	Zulassung zum Praktikum erfolgt bei bestandener Modulprüfung in Programmieren I oder Programmieren II
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit im Praktikum 90 Stunden Selbststudium
<b>Lehrformen:</b>	2 SWS seminaristischer Unterricht und Übungen 2 SWS Praktikum in kleinen Gruppen (14tägig 4 Stunden)
<b>Leistungsnachweise und Prüfung:</b>	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.
<b>Hinweise für dual Studierende:</b>	-

**Qualifikationsziele und Inhalte:****Qualifikationsziele:**

Die Studierenden kennen den Aufbau eines Betriebssystems und wichtige Konzepte, Probleme und Lösungen, die in einem Betriebssystem und bei der Entwicklung eines Betriebssystems zum Einsatz kommen. Sie sind in der Lage, einfache Betriebssystemfunktionalitäten zu entwickeln und bestehende Betriebssysteme zu verwalten, einzusetzen und zu beurteilen.

**Lehrinhalte:**

- Aufgaben und Aufbau von Betriebssystemen
- Prozesse und Threads: Nebenläufigkeit, Abstraktion, Prozessverwaltung, Prozesse und Threads in UNIX/Linux, POSIX Threads
- Synchronisation: elementare Synchronisationsbedingungen, komplexe Synchronisationsprobleme, Semaphoren, Monitore, Bedingungsvariablen, Fehlerquellen, Mechanismen in UNIX/Linux
- Interprozesskommunikation: Grundbegriffe, Shared Memory, Pipes, Message Queues, Sockets
- Scheduling: Scheduling-Strategien, Thread-Scheduling
- Speicherverwaltung: Speicherhierarchie, Speicherabstraktion, Swapping, Paging, Segmentierung, Seitenersetzungsalgorithmen, Realisierungsaspekte
- Dateisysteme: Dateien, Verzeichnisse, Implementierungsaspekte
- Eingabe und Ausgabe: I/O-Geräte, Adressierung, Schlüsselkonzepte, DMA, Schichten der I/O-Software
- Deadlocks: Erkennen und Beheben, Verhinderung, Vermeidung

**Literatur:**

A. Tanenbaum, H. Bos: Moderne Betriebssysteme, 4. Auflage, Pearson Studium, 2016

William Stallings: Operating Systems, 9. Auflage, Pearson Studium, 2018

Carsten Vogt: Nebenläufige Programmierung, Hanser Verlag, 2012

A. Silberschatz, P. B. Galvin, G. Gagne: Operating System Concepts, 8. Auflage, John Wiley and Sons Inc., 2009

## Datenkommunikation

IB420

<b>Modulverantwortlicher:</b>	<b>M. Sc. Thomas Franzke</b>
<b>Dozent:</b>	M. Sc. Lukas Reißmann
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im vierten Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	Erster Studienabschnitt oder vergleichbare Kenntnisse, Programmierkenntnisse
<b>Voraussetzungen:</b>	-
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit im Praktikum 90 Stunden Selbststudium
<b>Lehrformen:</b>	2 SWS seminaristischer Unterricht und Übungen 2 SWS Praktikum in kleinen Gruppen (14tägig 4 Stunden)
<b>Leistungsnachweise und Prüfung:</b>	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.
<b>Hinweise für dual Studierende:</b>	-

**Qualifikationsziele und Inhalte:****Qualifikationsziele:**

Die Studierenden besitzen Kenntnisse in Bereich der Netzwerktechnologien und -topologien, insbesondere über das Internet.

**Lehrinhalte:**

- OSI-Referenzmodell
- Grundlegende Mechanismen der Fluss- und Fehlerkontrolle
- Beschreibung und Codierung von Datentypen
- Serialisierung / Deserialisierung an Beispielen in verschiedenen Systemen
- Internet: Architektur, Protokolle, Anwendungen und Dienste
- Routing und Bridging
- Lokale Netze: z.B. Ethernet, WLAN
- Grundlagen der Übertragungstechniken und Fehlerkontrolle

**Literatur:**

Fred Halsall: Computer Networking and the Internet, Addison-Wesley 2005  
 K.R. Fall, W. R. Stevens: TCP/IP Illustrated, Vol. 1: The Protocols, Addison-Wesley 2011  
 U. Trick, F. Weber: SIP, TCP/IP und Telekommunikationsnetze, 2. Auflage, Oldenbourg 2009

## Statistik

IB430

<b>Modulverantwortlicher:</b>	<b>Prof. Dr. Eduard Kromer</b>
<b>Dozent:</b>	Prof. Dr. Sandra Eisenreich
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im vierten Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	Mathematik I und II oder vergleichbare Kenntnisse
<b>Voraussetzungen:</b>	
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	45 Stunden Präsenzzeit im Unterricht 15 Stunden Präsenzzeit im Praktikum 90 Stunden Selbststudium
<b>Lehrformen:</b>	3 SWS seminaristischer Unterricht und Übungen 1 SWS Praktikum (14 tägig, 2 SWS)
<b>Leistungsnachweise und Prüfung:</b>	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.
<b>Hinweise für dual Studierende:</b>	-

**Qualifikationsziele und Inhalte:****Qualifikationsziele:**

Die Studierenden haben Kenntnisse in den Grundlagen der Wahrscheinlichkeitstheorie und der Statistik erworben soweit diese für die Problemlösung von Aufgaben der Informatik benötigt werden. Der Einsatz statistischer Methoden bei der Lösung konkreter Fragestellungen wurde eingeübt. Die Studierenden kennen wichtige Anwendungen der Statistik in der Informatik.

**Lehrinhalte:**

Kombinatorik, Diskrete Wahrscheinlichkeitsräume, Bedingte Wahrscheinlichkeit, Zufallsvariablen, Stochastische Unabhängigkeit, Erwartungswert und Varianz, Kovarianz und Korrelation, Gesetz großer Zahlen, Zentraler Grenzwertsatz, Lage- und Streuungsparameter, Wichtige Verteilungen, Schätztheorie, Testtheorie.

**Literatur:**

Hartmann, Peter; Mathematik für Informatiker, Springer-Vieweg; 7. Auflage; 2019  
Georgii, Hans-Otto; Stochastik; de Gruyter, 5. Auflage; 2015  
Krengel, Ulrich; Einführung in die Wahrscheinlichkeitstheorie und Statistik; Springer-Vieweg; 8. Auflage; 2005  
Henze, Norbert; Stochastik für Einsteiger; Springer; 10. Auflage; 2013  
Meintrup, David; Schäffler, Stefan; Stochastik; Springer; 1. Auflage; 2005  
Behrends, Ehrhard; Elementare Stochastik; Springer-Vieweg; 2013

## Präsentation und Kommunikation

IB440

<b>Modulverantwortlicher:</b>	<b>M.Sc. Thomas Franzke</b>
<b>Dozent:</b>	Bernadette Jochum M.A.
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im Winter- und Sommersemester, max. 35 Plätze
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	-
<b>Voraussetzungen:</b>	-
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	60 Stunden Präsenzzeit im Unterricht 90 Stunden Selbststudium
<b>Lehrformen:</b>	4 SWS seminaristischer Unterricht und Übungen Eine Blockveranstaltung "Wissenschaftliches Arbeiten"
<b>Leistungsnachweise und Prüfung:</b>	Einzel-, Gruppenpräsentation und benotete Abschlusspräsentation
<b>Hinweise für dual Studierende:</b>	-

**Qualifikationsziele und Inhalte:****Qualifikationsziele:**

Die Teilnehmer sensibilisieren sich für ihren eigenen Kommunikationsstil, die eigene Körpersprache, lernen sich selbst und ihre Wirkung nach außen kennen und sind in der Lage, Fähigkeiten, Ideen und Erfolge überzeugend und authentisch zu kommunizieren. Die Teilnehmer erlernen wichtige Regeln für freies Reden und für die Gestaltung von Präsentationen; erfahren, wie sie mit Redehemmungen, Ängsten, Lampenfieber und Blackouts umgehen. Sie erleben sich in unterschiedlichen Kommunikationssituationen, initiieren Gespräche und bringen sie ziel- und dialogorientiert zum erfolgreichen Abschluss. Die Teilnehmer gewinnen Sicherheit im Auftreten vor Gruppen. Sie beherrschen die rhetorisch wirkungsvolle Präsentation unter Einsatz von Visualisierungshilfen. Sie gewinnen (begeistern) einen Zuhörerkreis oder Gesprächspartner für ihre Ideen und Zielvorstellungen.

Die Studierenden sind in der Lage, selbständig wissenschaftliche Arbeiten zu verfassen.

**Lehrinhalte:**

Einführung in die Grundlagen der Kommunikation. Bedeutung von Stimme Sprache, Körpersprache. Selbst- und Fremdwahrnehmung. Sensibilisierung für eigene Entwicklungsprozesse. Biografisches Arbeiten, Kompetenz- und Potentialanalyse.

Einzel-, Gruppenpräsentationen, Führen von berufsrelevanten Entscheidungsgesprächen, Themen aus Assessment u.a. Stressinterview.

Architektur von Präsentationen. Zielgruppenanalyse, Faktoren für Verständlichkeit und Überzeugungskraft, rhetorische Stilmittel, Umgang mit Nervosität. Motivation der Zuhörer, Positives Denken, Umgang mit Fragen und Störungen, schwierigen Gesprächspartnern.

Übungen können auch per Video-Analyse ausgewertet werden, alle Teilnehmer können individuell gecoacht werden.

Methoden zum Verfassen einer wissenschaftlichen Arbeit bzgl. Inhalt, Stil und Form.



**Literatur:**

- [1] Friedemann Schulz von Thun: „Miteinander reden: 1“ Allgemeine Psychologie der Kommunikation.
- [2] Vigenschow: Softskills für Softwareentwickler
- [3] Harry Holzheu: „Natürliche Rhetorik“
- [4] Julius Loewenstein: „Reden wie die Profis“
- [5] Martin Kornmeier; Wissenschaftlich schreiben leicht gemacht; Haupt Verlag; 6., aktuelle Auflage

## Praktische Zeit im Betrieb

IB500

<b>Modulverantwortlicher:</b>	<b>M.Sc. Thomas Franzke</b>
<b>Dozent:</b>	-
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im fünften Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	-
<b>Voraussetzungen:</b>	Erfolgreiches Bestehen aller Prüfungen der ersten beiden Studiensemester
<b>Leistungspunkte:</b>	22 (bei Ableistung im Ausland 28)
<b>Arbeitsaufwand:</b>	80 Arbeitstage Präsenzzeit im Betrieb
<b>Lehrformen:</b>	Tätigkeit in der Wirtschaft
<b>Leistungsnachweise und Prüfung:</b>	Praktikumsbericht in Textform (Benotung: mit/ohne Erfolg)
<b>Hinweise für dual Studierende:</b>	-

**Qualifikationsziele und Inhalte:****Qualifikationsziele:**

Die Studierenden haben vertiefte Kenntnisse über technische und organisatorische Problemlösungen in Betrieben.

**Lehrinhalte:**

Die Studierenden werden zum selbständigen und eigenverantwortlichen Arbeiten in praxisrelevanten DV-Projekten angeleitet. Die Mitarbeit sollte möglichst alle DV-Projektphasen, d.h.

- Systemanalyse
- Systemplanung
- Implementierung
- Systemeinführung

abdecken.

**Literatur:**

Tätigkeitsspezifisch

## Praxisseminar

IB510

<b>Modulverantwortlicher:</b>	<b>Prof. Dr. Christopher Auer</b>
<b>Dozent:</b>	Prof. Dr. Christopher Auer
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im fünften Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	-
<b>Voraussetzungen:</b>	Erfolgreiches Bestehen aller Prüfungen des ersten Studienabschnitts IB500 Muss parallel zu IB510 belegt werden oder bereits abgeleistet sein
<b>Leistungspunkte:</b>	3
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit im Unterricht 60 Stunden Selbststudium
<b>Lehrformen:</b>	2 SWS Seminar mit Kurzreferaten und Diskussion
<b>Leistungsnachweise und Prüfung:</b>	Teilnahmepflicht, benoteter Vortrag über das Praktikum IB500
<b>Hinweise für dual Studierende:</b>	-

**Qualifikationsziele und Inhalte:****Qualifikationsziele:**

Die Studierenden lernen verschiedene Felder der Informatik in der beruflichen, außeruniversitären Praxis kennen. Sie können ein umfangreiches Projekt verständlich und wohlstrukturiert präsentieren.

**Lehrinhalte:**

- Erfahrungsaustausch
- Anleitung und Beratung
- Fachliche Diskussion
- Präsentationsstil

**Literatur:**

Tätigkeitsspezifisch

## IT-Recht

IB530

<b>Modulverantwortlicher:</b>	<b>M.Sc. Thomas Franzke</b>
<b>Dozent:</b>	Andrea Schweizer-Zollner
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im fünften Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	-
<b>Voraussetzungen:</b>	-
<b>Leistungspunkte:</b>	3
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit im Unterricht 45 Stunden Selbststudium
<b>Lehrformen:</b>	2 SWS seminaristischer Unterricht (4 mal 7,5 Stunden)
<b>Leistungsnachweise und Prüfung:</b>	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.
<b>Hinweise für dual Studierende:</b>	-

**Qualifikationsziele und Inhalte:****Qualifikationsziele:**

Die Teilnehmer sind in der Lage, die wichtigsten Fallgestaltungen, die dem Rechtsanwender, Berater und Entscheider in der betrieblichen oder behördlichen IT-Praxis begegnen, eigenständig zu beurteilen und einer praxisgerechten, rechtssicheren Lösung zuzuführen bzw. rechtliche Problemlagen frühzeitig zu erkennen und Haftungsrisiken zu vermeiden.

**Lehrinhalte:**

Das IT-Recht ist keine geschlossene Rechtsmaterie wie etwa das Bürgerliche Recht. Es versteht sich vielmehr als Klammer um jene Ausschnitte der Rechtsordnung (Datenschutzrecht, Urheberrecht, Markenrecht, Strafrecht, Arbeitsrecht usw.), die auf Sachverhalte mit IT-Bezug anzuwenden sind. Insoweit ist das anzuwendende Recht ebenso vielfältig und komplex wie die zu beurteilenden Informations- und Kommunikationsvorgänge. Die gegenständliche Vorlesung soll dazu beitragen, die Herausforderungen der „Querschnittsmaterie“ IT-Recht zu meistern. Im Rahmen der Vorlesung werden in allgemein verständlicher Form diejenigen Rechtsfragen behandelt, die bei der Nutzung der „Neuen Medien“, insbesondere dem Internet, auftreten. Dabei werden gleichermaßen die zivilrechtlichen, die strafrechtlichen und die öffentlichrechtlichen Aspekte des Themas beleuchtet. Für alle Teilbereiche des Rechts sind insoweit fundierte Grundkenntnisse erforderlich, die in der Veranstaltung vermittelt werden.

Im Besonderen werden behandelt: Grundzüge des Datenschutzrechts, Recht der Telemedien, Domainrecht, Urheberrecht (Nutzung fremden Contents, File-Sharing, Rechte an Software usw.), Recht des E-Commerce (Verträge im Internet, der rechtskonforme Webshop, Online-Auktionen usw.), Recht des E-Government, Grundzüge des elektronischen Rechtsverkehrs, Telekommunikation am Arbeitsplatz (Nutzung und Kontrolle der Telekommunikation am Arbeitsplatz, Kündigung usw.)

Arbeitsmittel und zur Prüfung zugelassenes Hilfsmittel:

- Computerrecht, IT- und Computerrecht (Gesetzestexte), 7. Auflage, DTV-Ausgabe

**Literatur:**

Haug, Grundwissen Internetrecht, 2005  
 Köhler/Arndt/Fetzer, Recht des Internet, 6. Aufl. 2008  
 Steckler, Grundzüge des IT-Rechts, 2. Aufl. 2005  
 Heckmann, juris Praxiskommentar Internetrecht, 2007

# Grundlagen modernes Projektmanagement

IB540

<b>Modulverantwortlicher:</b>	<b>M.Sc. Thomas Franzke</b>
<b>Dozent:</b>	Susanne Messerer, Diplom-Betriebswirtin (FH)
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im fünften Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	-
<b>Voraussetzungen:</b>	-
<b>Leistungspunkte:</b>	3
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit im Unterricht 60 Stunden Selbststudium
<b>Lehrformen:</b>	2 SWS seminaristischer Unterricht
<b>Leistungsnachweise und Prüfung:</b>	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.
<b>Hinweise für dual Studierende:</b>	-

## Qualifikationsziele und Inhalte:

### Qualifikationsziele:

Die Studierenden haben das Basiswissen für klassisches und agiles Projektmanagement erlernt. Sie kennen die grundlegenden Begriffe, Methoden, Prozesse und Rollen und wissen, wie ein Projekt im klassischen als auch im agilen Projektmanagement durchgeführt wird. Die Studierenden erkennen, wann klassisches und wann agiles Vorgehen sinnvoll ist und sind sensibilisiert, das Beste aus beiden Ansätzen zu einem hybriden Projektmanagement zu vereinen.

### Lehrinhalte:

- Allgemeine Grundlagen des Projektmanagements
- Klassisches Projektmanagement
- Agiles Projektmanagement
- Scrum
- Kanban
- Hybrides Projektmanagement
- Führung und Teamentwicklung

### Literatur:

GPM Deutsche Gesellschaft für Projektmanagement e. V.: Kompetenzbasiertes Projektmanagement (PM4), 1. Auflage 2019  
 Holger Timinger: Modernes Projektmanagement, 1. Auflage 2017

## Numerik

IB605

<b>Modulverantwortlicher:</b>	<b>Prof. Dr. Michael Sagraloff</b>
<b>Dozent:</b>	Prof. Dr. Michael Sagraloff
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im dritten Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	Mathematik I und II oder vergleichbare Kenntnisse
<b>Voraussetzungen:</b>	Zulassung zum Praktikum erfolgt bei bestandener Prüfung in Mathematik I oder Mathematik II
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	60 Stunden Präsenzzeit im Unterricht 60 Stunden Präsenzzeit im Praktikum 90 Stunden Selbststudium
<b>Lehrformen:</b>	2 SWS seminaristischer Unterricht 2 SWS Praktikum in kleinen Gruppen (14tägig 4 Stunden)
<b>Leistungsnachweise und Prüfung:</b>	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.
<b>Hinweise für dual Studierende:</b>	-

**Qualifikationsziele und Inhalte:****Qualifikationsziele:**

Die Studierenden kennen die numerischen Methoden und Verfahren der Mathematik, die für die Problemlösung von Aufgaben der Informatik benötigt werden. Sie haben die Fähigkeit numerischer Methoden bei der Lösung von Problemen einzusetzen. Sie kennen wichtige Anwendungen der numerischen Mathematik in der Informatik.

**Lehrinhalte:**

- Direkte und iterative Methoden zur numerischen Lösung von linearen Gleichungssystemen
- Satz von Banach und numerische Behandlung von nichtlinearen Gleichungssystemen
- Numerische Behandlung von Polynomen
- Polynomapproximation und Splineapproximation
- Standardverfahren zur numerischen Integration
- Einführung zur numerischen Lösung von Differentialgleichungen

**Literatur:**

Hartmann, Peter: Mathematik für Informatiker, Vieweg 2006.  
Huckle, Schneider: Numerik für Informatiker, Springer Verlag

## Compiler

IB610

<b>Modulverantwortlicher:</b>	<b>Prof. Dr. Markus Mock</b>
<b>Dozent:</b>	Prof. Dr. Markus Mock
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im siebten Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	Grundlagen der Informatik, Grundlagen der theoretischen Informatik, Algorithmen und Datenstrukturen, Programmieren, Systemnahe Programmierung oder vergleichbare Kenntnisse
<b>Voraussetzungen:</b>	-
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit im Praktikum 90 Stunden Selbststudium
<b>Lehrformen:</b>	2 SWS seminaristischer Unterricht 2 SWS Praktikum in kleinen Gruppen (14tägig 4 Stunden)
<b>Leistungsnachweise und Prüfung:</b>	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.
<b>Hinweise für dual Studierende:</b>	-

**Qualifikationsziele und Inhalte:****Qualifikationsziele:**

Die Studierenden kennen die wichtigsten Gebiete des Compilerbaus. Aufgaben und Phasen eines Compilers, sowie die Realisierungsmöglichkeiten der in einem Übersetzungsprozess auftretenden Aufgaben sind bekannt. Die Studierenden können Compiler für neue Sprachen entwickeln und die Verfahren der Compilertechnik bei in der Praxis häufig auftretenden Problemstellungen anwenden.

**Lehrinhalte:**

- Überblick über die Compilerphasen
- Lexikalische Analyse
- Syntaxanalyseverfahren, Bottom-Up-Syntaxanalyse (LR), Top-Down-Syntaxanalyse (Rekursiver Abstieg, LL)
- Syntaxgesteuerte Übersetzung und attributierte Grammatiken
- Semantische Überprüfung
- Zwischencodegenerierung
- Codegenerierung
- Speicherverwaltung aus Compilersicht
- Registerallokation
- Instruktionsauswahl
- Code Scheduling
- Codeoptimierung

**Literatur:**

Aho, Lam, Sethi, Ullman: Compiler: Prinzipien, Techniken und Werkzeuge, Pearson Studium-IT 2008  
Steven Muchnick: Advanced Compiler Design and Implementation  
Bob Morgan: Building an Optimizing Compiler



## Verteilte Systeme

IB630

<b>Modulverantwortlicher:</b>	<b>Prof. Dr. Markus Mock</b>
<b>Dozent:</b>	Prof. Dr. Markus Mock
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	sechstes Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	Erster Studienabschnitt oder vergleichbare Kenntnisse, Grundkenntnisse in C, C++ und Java
<b>Voraussetzungen:</b>	-
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit im Praktikum 90 Stunden Selbststudium
<b>Lehrformen:</b>	2 SWS seminaristischer Unterricht 2 SWS Praktikum in kleinen Gruppen (14tägig 4 Stunden)
<b>Leistungsnachweise und Prüfung:</b>	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.
<b>Hinweise für dual Studierende:</b>	-

**Qualifikationsziele und Inhalte:****Qualifikationsziele:**

Die Studierenden besitzen grundlegende Kenntnisse im Bereich der Verteilung bei Software-Systemen. Sie können ausgewählte Frameworks für verteilte Systeme für die Programmierung verteilter Anwendungen benutzen.

**Lehrinhalte:**

- Verteilte Software-Systeme: Remote Method Invocation (RMI) und Remote Procedure Call (RPC)
- Verteilte Objektsysteme: Interface Description Language, Namensdienst
- Verteilte Datenbanken und Transaktionen
- Verteilte Dateisysteme
- Systemalgorithmen in verteilten Systemen
- Replikation und Konsistenz in verteilten Systemen
- Synchronisation in verteilten Systemen

**Literatur:**

G. Coulouris, J. Dollimore, T. Kindberg: Verteilte Systeme: Konzepte und Design, Addison-Wesley 2005  
A. Tanenbaum, M. v. Steen: Verteilte Systeme: Grundlagen und Paradigmen, Addison-Wesley 2007

# Internettechnologien

# IB640

<b>Modulverantwortlicher:</b>	<b>M.Sc. Markus Schmidtner</b>
<b>Dozent:</b>	M.Sc. Markus Schmidtner
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im sechsten Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	Erster Studienabschnitt oder vergleichbare Kenntnisse
<b>Voraussetzungen:</b>	-
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	60 Stunden Präsenzzeit im Unterricht 90 Stunden Selbststudium
<b>Lehrformen:</b>	2 SWS seminaristischer Unterricht 2 SWS Praktikum in kleinen Gruppen (14tägig 4 Stunden)
<b>Leistungsnachweise und Prüfung:</b>	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.
<b>Hinweise für dual Studierende:</b>	-

## Qualifikationsziele und Inhalte:

### Qualifikationsziele:

Die Studierenden sind mit den grundlegenden Netzwerktechnologien und Standards, wie dem ISO/OSI-Referenzmodell sowie den Protokollen des TCP/IP-Protokollstacks vertraut. Ferner können sie Dienste und Protokolle des Internets, wie DNS und HTTP, selbstständig anwenden und in von ihnen geschriebene Software integrieren. Darüber hinaus sind die Studierenden in Lage eigene Nachrichtenformate und XML-Sprachen zu definieren und zu verwenden. Weiterhin sind die Studierenden in der Lage eigene Client- und Serverseitige Webapplikationen zu entwickeln. Außerdem sollen die Studierenden die Grundlagen Serviceorientierter Architekturen erklären können.

### Lehrinhalte:

- Grundlagen von Standards im Internet: Geschichte, Organisation, Gremien, Standards.
- Grundlagen von Computernetzwerken: ISO/OSI-Referenzmodell, Protokolle des TCP/IP-Stacks, DNS, HTTP.
- Markup Languages: Aufbau von SGML, XML, DTD und XSD
- Client- und Serverseitige Webapplikationen: HTML, CSS, Javascript und PHP
- Grundlagen der Serviceorientierten Architekturen (SOA)

### Literatur:

Rüdiger Schreiner: Computernetzwerke: von den Grundlagen zur Funktion und Anwendung, 7. Aufl., Hanser, 2019  
 Daniel Takai: Architektur für Websysteme: Serviceorientierte Architektur, Microservices, Domänengetriebener Entwurf, Hanser, 2017  
 Christoph Meinel, Maxim Asjoma: Die neuen digitale Welt verstehen: Internet un WWW für alle, Springer, 2021

## Seminar

IB651

<b>Modulverantwortlicher:</b>	<b>M.Sc. Thomas Franzke</b>
<b>Dozent:</b>	Dozenten der Fakultät Informatik
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im sechsten Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	Informatik-Kenntnisse aus den ersten fünf Semestern des Bachelor-Studiums oder vergleichbare Kenntnisse
<b>Voraussetzungen:</b>	-
<b>Leistungspunkte:</b>	3
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit im Unterricht 60 Stunden Selbststudium
<b>Lehrformen:</b>	60 Minuten fachliche Präsentationen durch die Studierenden und anschließende Diskussionen.
<b>Leistungsnachweise und Prüfung:</b>	Teilnahmepflicht, 1 benotete Präsentation.
<b>Hinweise für dual Studierende:</b>	-

**Qualifikationsziele und Inhalte:****Qualifikationsziele:**

Die Studierenden sind in der Lage sich ein komplexes fachliches Thema aus der Literatur selbstständig zu erarbeiten. Sie können das Thema in einem fachlichen Vortrag unter Zuhilfenahme moderner Medien präsentieren und mit einem technisch versierten Publikum eine Diskussion über die Präsentations-Inhalte führen.

**Lehrinhalte:**

Aktuelle Themen der Informatik

**Literatur:**

[1] Abhängig von den behandelten Themen

## Prozessrechentechnik

IB700

<b>Modulverantwortlicher:</b>	<b>Prof. Dr. Martin Pellkofer</b>
<b>Dozent:</b>	Prof. Dr. Martin Pellkofer
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	im siebten Studiensemester
<b>Dauer:</b>	ein Semester
<b>Vorkenntnisse:</b>	Erster Studienabschnitt oder vergleichbare Kenntnisse
<b>Voraussetzungen:</b>	Zulassung zum Praktikum erfolgt bei bestandener Modulprüfung in Programmieren I oder Programmieren II
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit im Praktikum 90 Stunden Selbststudium
<b>Lehrformen:</b>	2 SWS seminaristischer Unterricht 2 SWS Praktikum in kleinen Gruppen (14tägig 4 Stunden)
<b>Leistungsnachweise und Prüfung:</b>	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.
<b>Hinweise für dual Studierende:</b>	-

**Qualifikationsziele und Inhalte:****Qualifikationsziele:**

Die Studierenden kennen die Eigenschaften und Charakteristika, in denen sich eingebettete Systeme von IT-Systemen unterscheiden. Sie verfügen über das notwendige Wissen in Systemdynamik und Regelungstechnik, um Prozessrechner als digitale Filter und Regler einsetzen zu können. Die Studierenden kennen neben Prozessoren weitere Arten von Verarbeitungseinheiten mit ihren Vor- und Nachteilen. Die Studierenden können externe Geräte wie Sensoren oder Aktoren an einem eingebetteten Rechner anbinden.

**Lehrinhalte:**

- Unterschiede zwischen eingebetteten Systemen und IT-Systemen
- Aufbau eingebetteter Systeme
- Grundlagen in Systemdynamik und Regelungstechnik
- Realisierung einfacher digitaler Filter und Regler mit Prozessrechner
- Sensoren und Aktuatoren eingebetteter Systeme
- Analoge und digitale I/O, Pulsweitenmodulation
- Analog/Digital-Wandlung, Alias-Effekte, Nyquist-Shannon-Abtasttheorem
- Modellbasiertes Design: Motivation, Grundprinzipien, x-in-the-Loop-Simulationen
- Entwicklung eingebetteter Softwaresysteme: Optimierung hinsichtlich Ressourcenverbrauch, Zuverlässigkeit, Risiko, Funktionale Sicherheit, Reliability Engineering
- Programmierung eines Mikrocontrollers (MPC560\*B von NXP): Timed I/O, Interrupts, ADC, DMA
- Entwicklung einer kamerabasierten Querführung für ein Modellfahrzeug

**Literatur:**

- P. Marwedel: Eingebettete Systeme, Springer Verlag, Heidelberg, 2008  
K. Berns, B. Schürmann, M. Trapp: Eingebettete Systeme, Vieweg+Teubner, 1. Auflage, Wiesbaden, 2010  
T. Benra, W. A. Halang: Software-Entwicklung für Echtzeitsysteme, Springer, Heidelberg, 2009  
K. Berns, B. Schürmann, M. Trapp: Eingebettete Systeme, Vieweg+Teubner, 1. Auflage, Wiesbaden, 2010  
Handbücher der benutzten Hardware und Software

## Fachbezogenes Wahlpflichtmodul I - V

IB7xx

<b>Modulverantwortlicher:</b>	<b>M.Sc. Thomas Franzke</b>
<b>Dozent:</b>	Dozenten der Informatikstudiengänge
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch
<b>Angebot:</b>	Im dritten/vierten bzw. im sechsten/siebten Studiensemester. Vor Beginn des Sommersemesters wird eine Liste der angebotenen Fächer mit ihren Beschreibungen veröffentlicht.
<b>Dauer:</b>	je ein Semester
<b>Vorkenntnisse:</b>	Erster Studienabschnitt oder vergleichbare Kenntnisse
<b>Voraussetzungen:</b>	-
<b>Leistungspunkte:</b>	jeweils 5
<b>Arbeitsaufwand:</b>	jeweils 60 Stunden Präsenzzeit im Unterricht jeweils 90 Stunden Selbststudium
<b>Lehrformen:</b>	Modulspezifisch
<b>Leistungsnachweise und Prüfung:</b>	Leistungsnachweise und Prüfungen werden in den individuellen Modulbeschreibungen festgelegt.
<b>Hinweise für dual Studierende:</b>	-

### Qualifikationsziele und Inhalte:

#### Qualifikationsziele:

Die Studierenden sind mit ausgewählten, fachbezogenen Wissensgebieten oder erweiterten Fertigkeiten in speziellen Anwendungen, die der individuellen Vorbereitung auf die berufliche Praxis dienen, vertraut

#### Lehrinhalte:

Siehe individuelle Fachbeschreibungen

#### Literatur:

Siehe individuelle Fachbeschreibungen

## Bachelor-Arbeit

IB720

<b>Modulverantwortlicher:</b>	<b>M.Sc. Thomas Franzke</b>
<b>Dozent:</b>	Hauptamtliche Dozenten der Fakultät Informatik
<b>Studiengang:</b>	Bachelor
<b>Modultyp:</b>	Pflichtfach
<b>Sprache:</b>	Deutsch oder Englisch
<b>Angebot:</b>	Nach Ableistung des praktischen Semesters
<b>Dauer:</b>	Fünf Monate
<b>Vorkenntnisse:</b>	-
<b>Voraussetzungen:</b>	Erfolgreiche Ableistung des praktischen Studiensemesters.
<b>Leistungspunkte:</b>	12
<b>Arbeitsaufwand:</b>	360 Stunden selbstständige Arbeit
<b>Lehrformen:</b>	Selbstständiges Arbeiten
<b>Leistungsnachweise und Prüfung:</b>	Schriftliche Bachelor-Arbeit, Kolloquium.
<b>Hinweise für dual Studierende:</b>	-

**Qualifikationsziele und Inhalte:****Qualifikationsziele:**

Die Studierenden können ein etwas größeres, aber zeitlich klar begrenztes, praxisbezogenes Informatik-Thema eigenständig und wissenschaftlich bearbeiten. Sie sind in der Lage, Problemstellungen und deren Lösungen schriftlich darzustellen und mündlich zu präsentieren.

**Lehrinhalte:**

Abhängig vom Thema der Arbeit

**Literatur:**

Abhängig vom Thema der Arbeit