

# Modulhandbuch

---

## Bachelor

# Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik

---

**Studienordnungsversion: 2013**

**Vertiefung: IN**

**gültig für das Sommersemester 2020**

Erstellt am: 01. Juli 2020

aus der POS Datenbank der TU Ilmenau

Herausgeber: Der Rektor der Technischen Universität Ilmenau

URN: urn:nbn:de:gbv:ilm1-mhb-17401

# Inhaltsverzeichnis

Name des Moduls/Fachs	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.F	Ab- schluss	LP
	V	S	P	V	S	P	V	S	P	V	S	P
<b>Informatik</b>											FP	60
Grundlagen und Diskrete Strukturen											FP	6
Grundlagen und Diskrete Strukturen			4 2 0								PL	6
Programmierparadigmen und Kommunikationsmodelle											FP	7
Kommunikationsmodelle				2 1 0							VL	3
Programmierparadigmen				2 2 0							VL	4
Softwaretechnik											FP	6
Softwaretechnik 1			2 1 0								PL 90min	3
Softwaretechnik 2				2 1 0							PL 90min	3
Telematik 1											FP	5
Telematik 1				3 1 0							PL 90min	5
Datenbank- und Betriebssysteme											FP	8
Betriebssysteme			2 1 0								VL	4
Datenbanksysteme			2 1 0								VL	4
Algorithmen, Automaten und Komplexität											FP	6
Algorithmen und Datenstrukturen 1			2 1 0								VL	3
Automaten und Komplexität				2 1 0							VL	3
Proseminar für IN Bsc											MO	2
Proseminar für IN Bsc					0 2 0						SL	2
Computergrafik											FP	5
Computergrafik					3 1 0						PL 60min	5
Logik und Logikprogrammierung											FP	5
Logik und Logikprogrammierung						3 2 0					PL 150min	5
Wahlpflichtmodul											FP	8
Rechnerarchitekturen für IN											FP	8
Rechnerarchitekturen 1				2 2 0							PL	4
Rechnerarchitekturen 2					2 1 0						PL	3
Praktikum Rechnerarchitekturen 1 und 2						0 0 1					SL	1
Neuroinformatik und Schaltsysteme											FP	8
Neuroinformatik				2 1 0							VL	3
Schaltsysteme				2 1 0							VL	3
Praktikum Neuroinformatik und Schaltsysteme					0 0 2						SL	2

---

## Modul: Informatik

Modulnummer: 101211

Modulverantwortlich: Prof. Dr. Martin Dietzfelbinger

Modulabschluss: Fachprüfung/Modulprüfung generiert

Lernergebnisse

siehe Fachbeschreibungen, für die im Studienplan zu diesem Modul angegebenen Fächer

Voraussetzungen für die Teilnahme

Detailangaben zum Abschluss

---

## Modul: Grundlagen und Diskrete Strukturen

Modulnummer: 100541

Modulverantwortlich: Prof. Dr. Matthias Kriesell

Modulabschluss: Fachprüfung/Modulprüfung generiert

Lernergebnisse

siehe Fachbeschreibung

Voraussetzungen für die Teilnahme

Detailangaben zum Abschluss

## Grundlagen und Diskrete Strukturen

Fachabschluss: mehrere Teilleistungen Art der Notengebung: Generierte Noten  
 Sprache: Deutsch/Englisch Pflichtkennz.: Pflichtfach Turnus: Wintersemester

Fachnummer: 100542 Prüfungsnummer: 240251

Fachverantwortlich: Prof. Dr. Matthias Kriesell

Leistungspunkte: 6 Workload (h): 180 Anteil Selbststudium (h): 112 SWS: 6.0  
 Fakultät für Mathematik und Naturwissenschaften Fachgebiet: 2411

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS					
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P
							4	2	0																								

### Lernergebnisse / Kompetenzen

Kenntnisse in Grundlagen der Mathematik, grundlegende mathematische Arbeitsweisen (Beweise)

### Vorkenntnisse

Abiturwissen Mathematik

### Inhalt

Logik und Mengenlehre; Funktionen und Relationen; Gruppen, Ringe, Körper; Boolesche Algebren; diskrete Wahrscheinlichkeitsräume; elementare Graphentheorie

### Medienformen

Tafel

### Literatur

Wird in der Vorlesung bekanntgegeben

### Detailangaben zum Abschluss

werden bei Bedarf festgelegt

### verwendet in folgenden Studiengängen:

- Bachelor Informatik 2013
- Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2013 Vertiefung
- Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2013 Vertiefung
- Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2013 Vertiefung
- Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2013 Vertiefung

## **Modul: Programmierparadigmen und Kommunikationsmodelle**

Modulnummer: 100333

Modulverantwortlich: Prof. Dr. Winfried Kühnhauser

Modulabschluss: Fachprüfung/Modulprüfung generiert

### **Lernergebnisse**

Die Studierenden lernen in diesem Modul grundlegende Paradigmen und Modelle zur Programmierung von Softwaresystemen und ihrer Kommunikation kennen. Sie verstehen die Grundlagen moderner Programmiersprachen und Middleware-Plattformen und sind in der Lage, sie bezüglich ihrer Leistung in unterschiedlichen Anwendungsdomänen der Informatik zu analysieren und bewerten.

### **Voraussetzungen für die Teilnahme**

### **Detailangaben zum Abschluss**

schriftliche Modulprüfung aus zwei Teilen (120 min)

## Kommunikationsmodelle

Fachabschluss: über Komplexprüfung

Art der Notengebung: unbenotet

Sprache: Deutsch

Pflichtkennz.: Pflichtfach

Turnus: Wintersemester

Fachnummer: 255

Prüfungsnummer: 2200378

Fachverantwortlich: Prof. Dr. Winfried Kühnhauser

Leistungspunkte: 3	Workload (h): 90	Anteil Selbststudium (h): 56	SWS: 3.0							
Fakultät für Informatik und Automatisierung			Fachgebiet: 2255							
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS
Fach-	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P
semester				2 1 0						

### Lernergebnisse / Kompetenzen

Die Fähigkeit der Kommunikation ist eine der grundlegenden Eigenschaften verteilter IT-Systeme. In diesem Kurs erwerben die Studierenden Grundlagenwissen über die zum Einsatz kommenden Kommunikationsmodelle in einem breiten Spektrum an Einsatzszenarien, beginnend bei eingebetteten verteilten Systemen bis hin zu globalen Informationssystemen. Sie lernen die grundlegenden Aufgaben, Funktionsweisen und Eigenschaften von Kommunikationsmodellen kennen, begreifen verteilte Systeme als strukturierte Systeme aus Komponenten mit individuellen Aufgaben und komplexen Kommunikationsbeziehungen und erwerben die Fähigkeit, problemspezifische Interaktionsmuster verteilter Systeme zu entwickeln und bezüglich ihrer Leistungen in unterschiedlichen Anwendungsdomänen zu analysieren und bewerten.

### Vorkenntnisse

Pflichtveranstaltung des Bachelor-Studiengangs Informatik des 1. Semesters

### Inhalt

Thematische Schwerpunkte sind

- Grundprinzipien, Paradigmen und Eigenschaften verteilter Systeme
- Kommunikationsmodelle (botschaftenbasierte/ ereignisbasierte/ strombasierte/ wissensbasierte/ auftragsorientierte/ funktionsaufruforientierte Modelle)
- Fallbeispiele, an denen die Anwendungen von Kommunikationsmodellen und Algorithmen in der Praxis veranschaulicht werden

### Medienformen

Präsentationen mit Projektor und Tafel, Bücher und Fachaufsätze, Übungsaufgaben und Diskussionsblätter

### Literatur

- George Coulouris et al.: Distributed Systems – Concepts and Design. 5th Ed., Addison-Wesley, 2012.
- Andrew S. Tanenbaum et al.: Distributed Systems – Principles and Paradigms. 2nd Ed., Pearson / Prentice Hall, 2007.
- Michel Raynal: Communication and Agreement Abstractions for Fault-Tolerant Asynchronous Distributed Systems. 1st Ed., Morgan & Claypool Publishers, 2010.
- Ingo Melzer et al.: Service-orientierte Architekturen mit Web Services – Konzepte – Standards – Praxis. 4. Auflage, Spektrum Akademischer Verlag, 2010.

### Detailangaben zum Abschluss

Teil der Modulprüfung (60 min)

### verwendet in folgenden Studiengängen:

Bachelor Informatik 2010

Bachelor Informatik 2013

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2013 Vertiefung

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2013 Vertiefung

Master Mathematik und Wirtschaftsmathematik 2008

## Programmierparadigmen

Fachabschluss: über Komplexprüfung

Art der Notengebung: unbenotet

Sprache: Deutsch

Pflichtkennz.: Pflichtfach

Turnus: Sommersemester

Fachnummer: 5378

Prüfungsnummer: 2200377

Fachverantwortlich: Prof. Dr. Günter Schäfer

Leistungspunkte: 4	Workload (h): 120	Anteil Selbststudium (h): 75	SWS: 4.0
Fakultät für Informatik und Automatisierung		Fachgebiet: 2256	

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS					
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			
										2	2	0																					

### Lernergebnisse / Kompetenzen

Die Studierenden erwerben Basiswissen über Programmiersprachparadigmen, einschließlich der zugrunde liegenden Denk- und Verarbeitungsmodelle. Sie können Programmiersprachen und deren Konzepte nach wesentlichen Paradigmen klassifizieren (Fachkompetenz). Die Studierenden sind in der Lage, zu gegebenen Problemen geeignete Paradigmen kritisch auszuwählen. Sie können einfache Programme sowohl im funktionalen als auch im objektorientierten Programmierstil systematisch entwerfen und implementieren (Methodenkompetenz). Die Studierenden verstehen verschiedene Programmiersprachkonzepte im Kontext einer Programmiersprache (Systemkompetenz). Die Studierenden können erarbeitete Lösungen einfacher Programmieraufgaben in der Gruppe analysieren und bewerten (Sozialkompetenz).

### Vorkenntnisse

Algorithmen und Programmierung ( 1. Semester )

### Inhalt

Übersicht über behandelte Programmierparadigmen:

- Objektorientiertes Paradigma (Schwerpunkt)
- Funktionales Paradigma (Schwerpunkt)
- Nebenläufiges Paradigma (Schwerpunkt)
- Paralleles Paradigma
- Generisches Paradigma
- Aspektorientiertes Paradigma
- Generatives Paradigma

Demonstriert werden alle Schwerpunkt-Konzepte an der auf Java basierenden multi-paradigmatischen Sprache Scala.

### Medienformen

Präsentationen, Handouts

### Literatur

wird aktuell im Web veröffentlicht

### Detailangaben zum Abschluss

schriftliche Prüfung (60 min)

verwendet in folgenden Studiengängen:

Bachelor Informatik 2010

Bachelor Informatik 2013

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2008 Vertiefung

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2013 Vertiefung

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2008 Vertiefung

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2013 Vertiefung



## Modul: Softwaretechnik

Modulnummer: 100334

Modulverantwortlich: Prof. Dr. Armin Zimmermann

Modulabschluss: Fachprüfung/Modulprüfung generiert

### Lernergebnisse

**Fachkompetenz:** Die Studierenden erwerben grundlegendes Wissen über Vorgehens- und Prozessmodelle der Softwareentwicklung, sowie über deren Methodik und Basiskonzepte. Sie können größere Entwicklungsaufgaben strukturieren, Lösungsmuster erkennen und anwenden, und verstehen den Entwurf von der Anforderungsermittlung bis hin zur Implementierung.

**Methodenkompetenz:** Den Studierenden wird Entscheidungskompetenz hinsichtlich möglicher Prinzipien, Methoden und Werkzeuge des ingenieurmäßigen Softwareentwurfs vermittelt.

**Systemkompetenz:** Die Studierenden verstehen das grundlegende Zusammenwirken unterschiedlicher Softwareentwicklungsphasen; anwendungsorientierte Kompetenzen bezüglich Modellierungsfähigkeit und Systemdenken werden geschult.

**Sozialkompetenz:** Die Studierenden verfügen über Fähigkeiten zur entwicklungsbezogenen, effektiven Teamarbeit.

### Vorraussetzungen für die Teilnahme

### Detailangaben zum Abschluss

## Softwaretechnik 1

Fachabschluss: Prüfungsleistung schriftlich 90 min Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch Pflichtkennz.: Pflichtfach Turnus: Wintersemester

Fachnummer: 100533 Prüfungsnummer: 2200369

Fachverantwortlich: Prof. Dr. Armin Zimmermann

Leistungspunkte: 3	Workload (h):90	Anteil Selbststudium (h):56	SWS:3.0							
Fakultät für Informatik und Automatisierung			Fachgebiet:2236							
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS
Fach-	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P
semester			2 1 0							

### Lernergebnisse / Kompetenzen

**Fachkompetenz:** Die Studierenden erwerben grundlegendes Wissen über Vorgehens- und Prozessmodelle der Softwareentwicklung, sowie über deren Methodik und Basiskonzepte. Sie können größere Entwicklungsaufgaben strukturieren, Lösungsmuster erkennen und anwenden, und verstehen den Entwurf von der Anforderungsermittlung bis hin zur Implementierung.

**Methodenkompetenz:** Den Studierenden wird Entscheidungskompetenz hinsichtlich möglicher Prinzipien, Methoden und Werkzeuge des ingenieurmäßigen Softwareentwurfs vermittelt.

**Systemkompetenz:** Die Studierenden verstehen das grundlegende Zusammenwirken unterschiedlicher Softwareentwicklungsphasen; anwendungsorientierte Kompetenzen bezüglich Modellierungsfähigkeit und Systemdenken werden geschult.

**Sozialkompetenz:** Die Studierenden verfügen über Fähigkeiten zur entwicklungsbezogenen, effektiven Teamarbeit.

### Vorkenntnisse

Algorithmen und Programmierung

### Inhalt

In der Lehrveranstaltung werden grundlegende Methoden, Modelle und Vorgehensweisen der Softwaretechnik bzw. des Software Engineering erlernt. Vorrangig wird die objektorientierte Sichtweise betrachtet, und in den Übungen anhand praktischer Beispiele vertieft. Für Implementierungsbeispiele wird vor allem JAVA verwendet.

- Einführung
- Modellierungskonzepte
  - . Überblick Modellierung
  - . klassische Konzepte (funktional, datenorientiert, algorithmisch, zustandsorientiert)
  - . Grundlagen Objektorientierung
  - . Unified Modeling Language (UML)
- Analyse
  - . Anforderungsermittlung
  - . Glossar, Geschäftsprozesse, Use Cases, Akteure
  - . Objektorientierte Analyse und Systemmodellierung
  - . Dokumentation von Anforderungen, Pflichtenheft
- Entwurf
  - . Software-Architekturen
  - . Objektorientiertes Design
  - . Wiederverwendung (Design Patterns, Komponenten, Frameworks, Bibliotheken)
- Implementierung
  - . Konventionen und Werkzeuge
  - . Codegenerierung
  - . Testen
- Vorgehensmodelle
  - . Überblick, Wasserfall, Spiralmodell, V-Modell XT, RUP, XP
- Projektmanagement
  - . Projektplanung
  - . Projektdurchführung

## Medienformen

Vorlesungsfolien, auf den Webseiten verfügbar  
Übungsaufgaben, auf den Webseiten verfügbar

## Literatur

- Brügge, Dutoit: Objektorientierte Softwaretechnik. Pearson 2004
- Balzert: Lehrbuch der Software-Technik - Basiskonzepte und Requirements Engineering.
- sowie ergänzende Literatur, siehe Webseiten und Vorlesung

## Detailangaben zum Abschluss

schriftliche Prüfung

verwendet in folgenden Studiengängen:

Bachelor Informatik 2013

Bachelor Ingenieurinformatik 2013

Bachelor Mathematik 2013

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2013 Vertiefung

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2013 Vertiefung

Bachelor Wirtschaftsinformatik 2013

Bachelor Wirtschaftsinformatik 2015

Master Mathematik und Wirtschaftsmathematik 2013 Vertiefung AM

Master Mathematik und Wirtschaftsmathematik 2013 Vertiefung WM

Master Medientechnologie 2017

## Softwaretechnik 2

Fachabschluss: Prüfungsleistung schriftlich 90 min Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch Pflichtkennz.: Pflichtfach Turnus: Sommersemester

Fachnummer: 100564 Prüfungsnummer: 2200379

Fachverantwortlich: Dr. Detlef Streitferdt

Leistungspunkte: 3	Workload (h):90	Anteil Selbststudium (h):56	SWS:3.0							
Fakultät für Informatik und Automatisierung			Fachgebiet:223A							
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS
Fach-	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P
semester				2 1 0						

### Lernergebnisse / Kompetenzen

**Fachkompetenz:** Die Studierenden verfügen über anwendungsorientiertes Wissen zu Werkzeugen der Anforderungserhebung und –modellierung, der Prozessmodellierung und – anpassung, der Aufwandsschätzung, des Softwaretests, der Produktlinienentwicklung und der Wartung von Software.  
**Methodenkompetenz:** Die Studierenden kennen den methodischen Hintergrund zu den vorgestellten Werkzeugen / Verfahren und sind daher in der Lage auch neue Problemstellungen zu lösen. Sie können aus den vorgestellten Methoden jeweils die passenden auswählen.  
**Systemkompetenz:** Die Studierenden können die vorgestellten Methoden und Werkzeuge in Projekten unterschiedlicher Domänen anwenden.  
**Sozialkompetenz:** Die Studierenden kennen die Bedeutung und den Einfluss der erlernten Methoden und Werkzeuge innerhalb einer Firma. Sie können daher ihr jeweiliges Vorgehen und die Ergebnisse auf die Erfordernisse eines Projektes in einer Organisation abstimmen.

### Vorkenntnisse

Softwaretechnik 1

### Inhalt

Diese Vorlesung vertieft die Inhalte der Softwaretechnik. Durch den Anwendungsbezug und die vorgestellten Entwicklungswerkzeuge werden theoretische Kenntnisse umgesetzt. Die bekannten Phasen des Softwareentwicklungszyklus werden durch Themen vertieft, deren Bedeutung im industriellen Praxiseinsatz hoch ist.

- Requirements Engineering (RE) - Als eine der wichtigen Grundvoraussetzungen für hochwertige Systeme gilt die Requirements Engineering Phase. Die wichtigsten Technologien werden vorgestellt und eingesetzt.
  - Elicitation, Modeling, Validation/Verification
  - Goal-Oriented RE
  - Traceability
  - RE Tool Support
  
- Softwareprozessmodellierung - Nutzung und Anpassung von Entwicklungsprozessen mit zugehörigen Artefakten (z. B. Checklisten, Dokumentvorlagen, Werkzeugen, Rollenkonzept, ...). Je nach Anforderung, sollen einzelne oder ganze Prozesse erzeugt und effizient eingesetzt werden, um eine Entwicklergruppe bestmöglich zu unterstützen.
  - Modellierung von Softwareentwicklungsprozessen (Wiederverwendung von Methoden- / Prozessschritten)
  - Tailoring von SW-Entwicklungsprozessen
  
- Langlebige Systeme - Das Wissen um den Lebenszyklus von Softwaresystemen ist entscheidend für deren Entwicklung und zukünftigen Erfolg. Die geforderte Stabilität langlebiger Systeme (z. B. mehr als 30 Jahre) muss sich im Entwurf der Systeme wiederfinden.
  - Design for Stability
  - Reengineering
  - Refactoring

- SW Wartung, Wartbarkeit

- Automatisiertes Testen - Veränderungen in den Anforderungen oder auch Fehlerbereinigungen führen zu der Notwendigkeit das System erneut testen zu müssen. Hierbei sind automatisierte Testansätze hilfreich. Zum einen lassen sie Änderungen an Testmodellen zu, aus denen Testfälle generiert werden. Zum anderen können Testfälle mit unterschiedlichen Zielen generiert werden, z. B. der Verbesserung der Codeabdeckung.

- Einordnung in den SW-Entwicklungsprozess

- Testmodellierung

- Testfallableitung

- Analyse von Testergebnissen

- Software Produktlinien - Der immer häufiger angewandte Produktlinienansatz erfordert ein Umdenken während des gesamten Entwicklungszyklus. Sollen später Produkte generiert und nicht jeweils als Eigenentwicklung entstehen, sind folgende Themen relevant:

- Merkmalmodelle (variable / gemeinsame Systemanteile)

- Produktlinien Architekturen

- Domänenspezifische Sprachen

- Testen von Produktlinien

- Generieren von Applikationen aus einer Produktlinie

## Medienformen

Bücher, Webseiten, Wissenschaftliche Paper, Open Source/Kommerzielle – Werkzeuge

## Literatur

[Boec 2004] Günter Böckle, Peter Knauber, Klaus Pohl, Klaus Schmid, "Software-Produktlinien: Methoden, Einführung und Praxis", dpunkt.Verlag GmbH, 2004.

[Broo 1995] Frederick P. Brooks, Jr., "The Mythical Man Month", Addison-Wesley, 1995.

[Fowl 1999] Martin Fowler, "Refactorings – Improving the Design of Existing Code", Addison Wesley, 1999.

[Gamm 1995] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, "Design Pattern – Elements of Reusable Object-Oriented Software", Addison Wesley, 1995.

[Lams 2009] Axel van Lamsweerde, "Requirements Engineering: From System Goals to UML Models to Software Specifications", John Wiley & Sons, 2009.

[McCo 2006] Steve McConnell, "Software Estimation", Microsoft Press, 2006.

[Pohl 2005] Klaus Pohl, Günter Böckle, Frank van der Linden, "Software Product Line Engineering – Foundations, Principles, and Techniques", Springer, Heidelberg 2005.

[Pohl 2008] Klaus Pohl, "Requirements Engineering: Grundlagen, Prinzipien, Techniken", dpunkt.Verlag GmbH, 2008.

[Robe 1999] Suzanne Robertson, James Robertson, "Mastering the Requirements Process", Addison-Wesley, 1999.

[Rooc 2004] Stefan Roock, Martin Lippert, "Refactorings in großen Softwareprojekten", dpunkt.Verlag GmbH, 2004.

[Somm 2007] Ian Sommerville, "Software Engineering", Pearson Studium, 2007.

[Wieg 1999] Karl E. Wiegers, "Software Requirements", Microsoft Press, 1999.

[Your 1997] Edward Yourdon, "Death March", Prentice-Hall, 1997.

## Detailangaben zum Abschluss

Schriftliche Prüfung.

## verwendet in folgenden Studiengängen:

Bachelor Informatik 2013

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2013 Vertiefung

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2013 Vertiefung

Master Mathematik und Wirtschaftsmathematik 2013 Vertiefung AM

Master Mathematik und Wirtschaftsmathematik 2013 Vertiefung WM

---

## Modul: Telematik 1

Modulnummer: 100322

Modulverantwortlich: Prof. Dr. Günter Schäfer

Modulabschluss: Fachprüfung/Modulprüfung generiert

### Lernergebnisse

siehe Fachbeschreibung

### Vorraussetzungen für die Teilnahme

Hochschulzulassung;

Grundlagenvorlesung in Informatik oder Programmierung (z.B. „Algorithmen und Programmierung“ oder eine vergleichbare Grundlagenvorlesung)

### Detailangaben zum Abschluss

Schriftliche Prüfung

## Telematik 1

Fachabschluss: Prüfungsleistung schriftlich 90 min Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch Pflichtkennz.: Pflichtfach Turnus: Sommersemester

Fachnummer: 100575 Prüfungsnummer: 2200383

Fachverantwortlich: Prof. Dr. Günter Schäfer

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 105	SWS: 4.0							
Fakultät für Informatik und Automatisierung			Fachgebiet: 2253							
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS
Fach-	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P
semester				3 1 0						

### Lernergebnisse / Kompetenzen

- **Fachkompetenz:** Die Studierenden verfügen über Kenntnisse und Überblickswissen zu Aufbau und Funktionsweise von Netzen, insbesondere des Internet.
- **Methodenkompetenz:** Die Studierenden sind in der Lage, einfache Protokollfunktionen zu spezifizieren und in Programmfragmente umzusetzen. Sie können die Auswirkungen bestimmter Entwurfsentscheidungen bei der Realisierung einzelner Protokollfunktionen auf grundlegende Leistungskenngrößen einschätzen. Sie kennen Darstellung von Protokollabläufen in Form von Message Sequence Charts und können gültige Protokollabläufe auf der Grundlage von Zustandsautomaten nachvollziehen.
- **Systemkompetenz:** Die Studierenden verstehen das grundsätzliche Zusammenwirken der Komponenten eines Netzes als System.
- **Sozialkompetenz:** Die Studierenden erarbeiten Problemlösungen einfacher Protokollfunktionen (z.B. Routing, Fehlerkontrolle, Flusskontrolle etc.) durch Bearbeiten von Übungsaufgaben in Gruppen und vertiefen bei Behandlung des Themas Geteilter Medienzugriff die technische Motivation für die Vorteile einer koordinierten Zusammenarbeit.
- Die Studierenden verfügen über Kenntnisse und Überblickswissen zu den anwendungsorientierten Schichten von Netzen und deren Protokolle, insbesondere des Internet. Die Studierenden kennen die grundlegenden Sicherheitsanforderungen an Kommunikationsdienste und Mechanismen zu ihrer Erfüllung.

### Vorkenntnisse

Hochschulzulassung;  
 Grundlagenvorlesung in Informatik oder Programmierung (z.B. „Algorithmen und Programmierung“ oder eine vergleichbare Grundlagenvorlesung)

### Inhalt

1. Einführung und Überblick: Grundsätzlicher Netzaufbau; Protokollfunktionen; Spezifikation; Architektur; Standardisierung; OSI- und Internet-Architekturmodell
2. Physikalische Schicht: Begriffe: Information, Daten und Signale; Physikalische Eigenschaften von Übertragungskanälen (Dämpfung, Verzerrung, Rauschen); Grenzen erreichbarer Datenübertragungsraten (Nyquist, Shannon); Taktsynchronisation; Modulationsverfahren (Amplituden-, Frequenz- und Phasenmodulation, kombinierte Verfahren)
3. Sicherungsschicht: Rahmensynchronisation; Fehlererkennung (Parität, Checksummen, Cyclic Redundancy Code; Fehlerbehebung (Forward Error Correction, Automatic Repeat Request); ARQ-Protokolle: Stop and Wait, Go-Back-N, Selective Reject; Medienzugriffsverfahren (ALOHA, Slotted ALOHA, Token-Ring, CSMA/CD); Ethernet; Internetworking: Repeater, Brücken und Router
4. Netzwerkschicht: Virtuelle Verbindungen vs. Datagramme; Aufgaben, Funktion und Aufbau eines Routers; Internet Protocol (IP): Paketaufbau und Protokollfunktionen, Hilfsprotokolle und Protokollversionen; Routingalgorithmen: Distanzvektor- und Link-State-Verfahren; Routingprotokolle des Internet (RIP, OSPF, BGP)
5. Transportschicht: Adressierung und Multiplexing; Verbindungsloser vs. verbindungsorientierter Transportdienst; Fehlerkontrolle; Flusskontrolle; Staukontrolle; Transportprotokolle des Internet (TCP, UDP)
6. Anwendungsorientierte Schichten: Sitzungsschicht, Darstellungsschicht und Anwendungsschicht, Grundarchitekturen verteilter Anwendungen: Client-Server, Peer-to-Peer, hybride Ansätze, Konkrete Protokolle der Anwendungsschicht: HTTP, SMTP, DNS;
7. Netzsicherheit

### Medienformen

Vorlesung mit Tafel und Folien-Präsentationen, Arbeitsblätter, Lehrbuch

#### Literatur

A. S. Tanenbaum. Computernetzwerke. Pearson Education. · J. F. Kurose, K. W. Ross. Computernetze. Pearson Education.

#### Detailangaben zum Abschluss

#### verwendet in folgenden Studiengängen:

Bachelor Informatik 2013

Bachelor Ingenieurinformatik 2013

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2013 Vertiefung

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2013 Vertiefung

Master Mathematik und Wirtschaftsmathematik 2013 Vertiefung AM

Master Mathematik und Wirtschaftsmathematik 2013 Vertiefung WM



## Modul: Datenbank- und Betriebssysteme

Modulnummer: 100335

Modulverantwortlich: Prof. Dr. Winfried Kühnhauser

Modulabschluss: Fachprüfung/Modulprüfung generiert

### Lernergebnisse

Thema in diesem Modul sind die grundlegenden Aufgaben, Paradigmen, funktionalen und nichtfunktionalen Eigenschaften zweier elementarer Themenkomplexe in der Informatik: den Datenbank- und Betriebssystemen. Die Teilnehmer lernen Datenbank- und Betriebssysteme als strukturierte parallele Systeme aus Komponenten mit individuellen Aufgaben und hochgradig komplexen Beziehungen kennen. Sie lernen die theoretischen Grundlagen, Abstraktionen und Paradigmen kennen und erwerben Kenntnisse über Prinzipien, Methoden, Algorithmen und Datenstrukturen, mit denen funktionale und nichtfunktionale Eigenschaften von Datenbank- und Betriebssystemen realisiert werden.

Auf dieser Grundlage besitzen Studierende nach Abschluss dieses Moduls die Fähigkeit, Datenbank- und Betriebssysteme bezüglich ihrer Eignung und Leistungen in unterschiedlichen Anwendungsdomänen zu analysieren, zu bewerten und einzusetzen und kennen grundlegende Methoden und Verfahren zu ihrem Entwurf und ihrer Implementierung.

### Vorraussetzungen für die Teilnahme

### Detailangaben zum Abschluss

Schriftliche Prüfung

## Betriebssysteme

Fachabschluss: über Komplexprüfung

Art der Notengebung: unbenotet

Sprache: Deutsch

Pflichtkennz.: Pflichtfach

Turnus: Wintersemester

Fachnummer: 252

Prüfungsnummer: 2200322

Fachverantwortlich: Prof. Dr. Winfried Kühnhauser

Leistungspunkte: 4	Workload (h): 120	Anteil Selbststudium (h): 86	SWS: 3.0							
Fakultät für Informatik und Automatisierung			Fachgebiet: 2255							
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS
Fach-	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P
semester			2 1 0							

### Lernergebnisse / Kompetenzen

Betriebssysteme bilden das Software-Fundament aller informationstechnischen Systeme. Ihre funktionalen und vor allem ihre nichtfunktionalen Eigenschaften wie Robustheit, Sicherheit oder Effizienz üben einen massiven Einfluss auf sämtliche Softwaresysteme aus, die unter ihrer Kontrolle ablaufen.

Dieser Kurs vermittelt Wissen über die grundlegenden Aufgaben, Funktionen und Eigenschaften von Betriebssystemen. Er stellt ihre elementaren Abstraktionen und Paradigmen vor und erklärt Prinzipien, Algorithmen und Datenstrukturen, mit denen funktionale und nichtfunktionale Eigenschaften realisiert werden.

Die Kursteilnehmer lernen Betriebssysteme als strukturierte parallele Systeme aus Komponenten mit individuellen Aufgaben und hochgradig komplexen Beziehungen verstehen; sie erwerben die Fähigkeit, Betriebssysteme bezüglich ihrer Eignung und Leistungen in unterschiedlichen Anwendungsdomänen zu analysieren, zu bewerten und einzusetzen.

### Vorkenntnisse

Algorithmen und Programmierung, Algorithmen und Datenstrukturen, Rechnerarchitekturen, Programmier- und Kommunikationsparadigmen

### Inhalt

Kursinhalte sind

- Nebenläufigkeit und Parallelität, Prozess- und Threadmodelle, Scheduling, Synchronisation und Kommunikation
- Ressourcenmanagement, Prozessoren, virtueller Speicher, Kommunikation
- Dateisysteme
- Netzwerkmanagement
- Ein-/Ausgabesysteme
- Architekturprinzipien

### Medienformen

Präsentationen mit Projektor und Tafel, Bücher und Fachaufsätze, Übungsaufgaben und Diskussionsblätter

### Literatur

- Andrew S. Tanenbaum: Modern Operating Systems. Pearson / Prentice Hall.
- William Stallings: Operating Systems - Internals and Design Principles. Pearson / Prentice Hall.
- Brian L. Stuart: Principles of Operating Systems. Thomson Learning / Course Technology
- Gary Nutt: Operating Systems - A Modern Perspective. Addison-Wesley.
- Gadi Taubenfeld: Synchronization Algorithms and Concurrent Programming. Pearson / Prentice Hall.
- David Mosberger, Stephane Eranian: IA-64 Linux Kernel - Design and Implementation. Prentice Hall.
- Daniel P. Bovet, Marco Cesati: Understanding the Linux Kernel. O'Reilly & Associates.
- Jonathan Levin: Mac OS X and iOS Internals. John Wiley & Sons.

### Detailangaben zum Abschluss

schriftliche Modulprüfung

verwendet in folgenden Studiengängen:

Bachelor Informatik 2010

Bachelor Informatik 2013

Bachelor Ingenieurinformatik 2008  
Bachelor Ingenieurinformatik 2013  
Bachelor Mathematik 2009  
Bachelor Mathematik 2013  
Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2008 Vertiefung  
Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2013 Vertiefung  
Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2008 Vertiefung  
Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2013 Vertiefung  
Bachelor Wirtschaftsinformatik 2009  
Bachelor Wirtschaftsinformatik 2010  
Bachelor Wirtschaftsinformatik 2011  
Bachelor Wirtschaftsinformatik 2013  
Bachelor Wirtschaftsinformatik 2015  
Master Mathematik und Wirtschaftsmathematik 2008  
Master Mathematik und Wirtschaftsmathematik 2013 Vertiefung AM  
Master Mathematik und Wirtschaftsmathematik 2013 Vertiefung WM

## Datenbanksysteme

Fachabschluss: über Komplexprüfung

Art der Notengebung: unbenotet

Sprache: Deutsch

Pflichtkennz.: Pflichtfach

Turnus: Wintersemester

Fachnummer: 244

Prüfungsnummer: 2200323

Fachverantwortlich: Prof. Dr. Kai-Uwe Sattler

Leistungspunkte: 4	Workload (h): 120	Anteil Selbststudium (h): 86	SWS: 3.0																								
Fakultät für Informatik und Automatisierung		Fachgebiet: 2254																									
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS																	
Fach-	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P
semester																											

### Lernergebnisse / Kompetenzen

Nach dem Besuch dieser Veranstaltung können die Studierenden Datenbanksysteme anwenden. Sie kennen die Schritte des Entwurfs von Datenbanken und können die relationale Entwurfstheorie beschreiben. Weiterhin können sie deklarative Anfragen in SQL und XPath/XQuery formulieren sowie Integritätsbedingungen definieren. Die Studierenden sind in der Lage, gegebene praktische Problemstellungen zu analysieren, im ER-Modell zu modellieren und in einer relationalen Datenbank abzubilden sowie SQL zur Anfrageformulierung zu nutzen.

### Vorkenntnisse

Vorlesung Algorithmen und Programmierung

### Inhalt

Grundbegriffe von Datenbanksystemen; Phasen des Datenbankentwurfs, Datenbankentwurf im Entity-Relationship-Modell, Relationaler Datenbankentwurf, Entwurfstheorie, Funktionale Abhängigkeiten und Normalformen; Grundlagen von Anfragen: Algebra und Kalküle; SQL: relationaler Kern und Erweiterungen, rekursive Anfragen mit SQL; Transaktionen und Integritätssicherung; Sichten und Zugriffskontrolle; XPath & XQuery als Anfragesprachen für XML

### Medienformen

Vorlesung mit Präsentation und Tafel, Handouts, Moodle

### Literatur

Saake, Sattler, Heuer: Datenbanken – Konzepte und Sprachen, 4. Auflage, mitp-Verlag, 2010.

### Detailangaben zum Abschluss

schriftliche Modulprüfung

verwendet in folgenden Studiengängen:

Bachelor Informatik 2013  
 Bachelor Ingenieurinformatik 2008  
 Bachelor Ingenieurinformatik 2013  
 Bachelor Mathematik 2013  
 Bachelor Medientechnologie 2008  
 Bachelor Medientechnologie 2013  
 Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2013 Vertiefung  
 Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2013 Vertiefung  
 Bachelor Wirtschaftsinformatik 2009  
 Bachelor Wirtschaftsinformatik 2010  
 Bachelor Wirtschaftsinformatik 2011  
 Bachelor Wirtschaftsinformatik 2013  
 Bachelor Wirtschaftsinformatik 2015  
 Master Mathematik und Wirtschaftsmathematik 2013 Vertiefung AM  
 Master Mathematik und Wirtschaftsmathematik 2013 Vertiefung WM

## Modul: Algorithmen, Automaten und Komplexität

Modulnummer: 100321

Modulverantwortlich: Prof. Dr. Dietrich Kuske

Modulabschluss: Fachprüfung/Modulprüfung generiert

### Lernergebnisse

Die Studierenden

- können grundlegende Algorithmen und Datenstrukturen anhand von Beispielen beschreiben,
- sind in der Lage, die Effizienz von Algorithmen zu bestimmen und
- können geeignete Algorithmen auswählen bzw. anpassen.

Die Studierenden

- kennen die Stufen 0,2 und 3 der Chomsky-Hierarchie, ihre automatentheoretischen Charakterisierungen und können die Umwandlungen ausführen,
- sind in der Lage, Nachweise der Nicht-Regularität zu führen,
- sind mit Abschluß- und algorithmischen Eigenschaften der regulären und kontextfreien Sprachen vertraut.

Die Studierenden

- kennen die Klassen der semi-entscheidbaren und der entscheidbaren Probleme mit wesentlichen Beispielen,
- sind in der Lage, die Church-Turing-These zu formulieren, ihre Bedeutung darzustellen und Argumente zu ihrer Begründung anzugeben.

Die Studierenden

- kennen die Komplexitätsklassen P und NP und einige NP-vollständige Probleme,
- können erläutern, warum die NP-schweren Probleme nicht effizient lösbar sind.

### Voraussetzungen für die Teilnahme

### Detailangaben zum Abschluss

# Algorithmen und Datenstrukturen 1

Fachabschluss: über Komplexprüfung

Art der Notengebung: unbenotet

Sprache: Deutsch

Pflichtkennz.: Pflichtfach

Turnus: Sommersemester

Fachnummer: 100534 Prüfungsnummer: 2200370

Fachverantwortlich: Prof. Dr. Martin Dietzfelbinger

Leistungspunkte: 3	Workload (h): 90	Anteil Selbststudium (h): 56	SWS: 3.0
Fakultät für Informatik und Automatisierung			Fachgebiet: 2242

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS					
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			
							2	1	0																								

## Lernergebnisse / Kompetenzen

Die Studierenden kennen beispielhaft Methoden für die Spezifikation von Berechnungsproblemen und von abstrakten Datentypen und können sie für die grundlegenden Aufgaben und Datentypen anwenden. Sie können unter Verwendung der O-Notation die Laufzeit von (auch zusammengesetzten) iterativen und rekursiven Algorithmen analysieren. Sie können auf iterative und rekursive Algorithmen Induktionsargumente anwenden, um die Korrektheit zu zeigen. Die Studierenden kennen die grundlegenden Datenstrukturen „Array“, „Liste“, „Stack“, „Queue“, „gerichteter Baum“ und „Binärbaum“ mit ihren Implementierungsmöglichkeiten und können die zentralen Performanzparameter benennen und begründen. Sie kennen den Datentyp „binärer Suchbaum“ mit seinen Methoden für Einfügen und Suchen und den Datentyp „Mehrwegsuchbaum“. Die Studierenden kennen die Algorithmen für eine Variante von balancierten Binärbäumen und können sie an Beispielen durchführen. Die Studierenden kennen das Prinzip von einfachen Hashverfahren, verstehen die Funktionsweise und können das zu erwartende Verhalten für die verschiedenen Verfahren beschreiben. Sie kennen Konstruktionen einfacher randomisierter Hashklassen. Die Studierenden kennen die grundlegenden Sortieralgorithmen (Quicksort, Heapsort, Mergesort sowie Radixsort), können die Korrektheit der Verfahren begründen und ihre Laufzeit berechnen. Sie kennen den grundlegenden Datentyp „Priority Queue“ und seine Implementierung auf der Basis von binären Heaps. Die Studierenden kennen die Grundbegriffe der Graphentheorie und können mit ihnen umgehen. Sie kennen die wesentlichen Datenstrukturen für die Darstellung von Graphen und Digraphen mit den zugehörigen Methoden und Performanzparametern, sowie einfache Graphdurchmusterungsverfahren (Breitensuche, Tiefensuche). Sie kennen einen Algorithmus für die Ermittlung eines minimalen Spannbaums. Sie können die Korrektheit dieses Verfahrens begründen.

## Vorkenntnisse

Algorithmen und Programmierung, Mathematik I.

## Inhalt

Spezifikation von Berechnungsproblemen und von abstrakten Datentypen. Analyse von Algorithmen: Korrektheitsbeweise für iterative und rekursive Verfahren, Laufzeitbegriff, O-Notation, Laufzeitanalyse. Grundlegende Datenstrukturen (Listen, Stacks, Queues, Bäume). Binäre Suchbäume, Mehrwegsuchbäume, balancierte Suchbäume (AVL- und/oder Rot-Schwarz-Bäume, B-Bäume). Einfache Hashverfahren, universelles Hashing. Sortierverfahren: Quicksort, Heapsort, Mergesort, Radixsort. Priority Queues mit der Implementierung als Binärheaps. Grundbegriffe der Graphentheorie (ungerichtete und gerichtete Graphen, Markierungen an Knoten und Kanten, Wege und Kreise, Bäume und Wälder, Zusammenhangskomponenten). Datenstrukturen für Graphen (Adjazenzmatrix, Kantenliste, Adjazenzlisten, Adjazenzarrays). Durchmustern von Graphen: Breitensuche, Tiefensuche, Zusammenhangskomponenten, Entdecken von Kreisen. Minimale Spannbaume. Begleitend: Methoden für die Analyse von Laufzeit und Korrektheit.

## Medienformen

Projektion von Folien.

## Literatur

- \* G. Saake, K.-U. Sattler, Algorithmen und Datenstrukturen, 4. Auflage, dpunkt, 2010.
- \* R. Sedgewick, Algorithmen in C++, Pearson Studium, 2001.
- \* T. Ottmann, P. Widmayer, Algorithmen und Datenstrukturen, Spektrum Akademischer Verlag, 4., überarb. Aufl. 2002
- \* R. H. Güting, S. Dieker: Datenstrukturen und Algorithmen, Teubner, 2004

\* T. Cormen, C. Leiserson, R. Rivest, C. Stein, Introduction to Algorithms, Second Edition, MIT Press 2001.

(Auch auf deutsch erhältlich.)

\* M. T. Goodrich, R. Tamassia, Data Structures and Algorithms in Java, 2. Auflage, Wiley, 2003.

Detailangaben zum Abschluss

verwendet in folgenden Studiengängen:

Bachelor Ingenieurinformatik 2013

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2013 Vertiefung

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2013 Vertiefung

## Automaten und Komplexität

Fachabschluss: über Komplexprüfung

Art der Notengebung: unbenotet

Sprache: Deutsch

Pflichtkenn.: Pflichtfach

Turnus: Sommersemester

Fachnummer: 100535

Prüfungsnummer: 2200371

Fachverantwortlich: Prof. Dr. Dietrich Kuske

Leistungspunkte: 3	Workload (h):90	Anteil Selbststudium (h):56	SWS:3.0																								
Fakultät für Informatik und Automatisierung			Fachgebiet:2241																								
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS																	
Fach-	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P
semester							2	1	0																		

### Lernergebnisse / Kompetenzen

#### Die Studierenden

- kennen die Stufen 0, 2 und 3 der Chomsky-Hierarchie, ihre automatentheoretischen Charakterisierungen und können die Umwandlungen ausführen,
- sind in der Lage, Nachweise der Nicht-Regularität zu führen,
- sind mit Abschluss- und algorithmischen Eigenschaften der regulären und kontextfreien Sprachen vertraut.

#### Die Studierenden

- kennen die Klassen der semi-entscheidbaren und der entscheidbaren Probleme mit wesentlichen Beispielen,
- sind in der Lage, die Church-Turing These zu formulieren, ihre Bedeutung darzustellen und Argumente zu ihrer Begründung anzugeben.

#### Die Studierenden

- kennen die Komplexitätsklassen P und NP und einige NP-vollständige Probleme,
- können erläutern, warum die NP-schweren Probleme nicht effizient lösbar sind.

### Vorkenntnisse

Umgang mit mengentheoretischen Begriffen und Notationen (z.B. erworben in "Mathematik 1")

### Inhalt

- reguläre Sprachen: deterministische und nichtdeterministische endliche Automaten, reguläre Ausdrücke, effektiver Abschluss unter Booleschen Operationen, Verkettung und Iteration, Entscheidbarkeit von Leerheit, Inklusion und Äquivalenz, Pumping-Lemma
- kontextfreie Sprachen: Kellerautomaten, Parsing, effektiver Abschluss unter positiven Booleschen Operationen, Verkettung und Iteration, Entscheidbarkeit der Leerheit, Pumping-Lemma, Nichtabschluss unter Schnitt
  - semi-entscheidbare Sprachen: Turing-Maschine
  - Berechenbarkeit: Church-Turing These, Unentscheidbarkeit des Halteproblems und der Universalität von kontextfreien Sprachen
  - Komplexitätstheorie: die Klassen P und NP, NP-Vollständigkeit, 3-SAT, graphentheoretische Probleme

### Medienformen

Tafel, Übungsblätter

### Literatur

Schöning "Theoretische Informatik kurzgefasst"

### Detailangaben zum Abschluss

verwendet in folgenden Studiengängen:

Bachelor Ingenieurinformatik 2013

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2013 Vertiefung





## Modul: Proseminar für IN Bsc

Modulnummer: 100346

Modulverantwortlich: Prof. Dr. Winfried Kühnhauser

Modulabschluss:

### Lernergebnisse

Ein Proseminar im Bachelorstudium dient dem Erlernen und Einüben des Umgangs mit technisch/wissenschaftlichen Texten. Hauptaufgabe eines Studierenden ist die eigenständige, aber betreute Erarbeitung wissenschaftlicher Literatur bis zum eigenen Verständnis und die geschlossene Darstellung dieses Materials in einem Vortrag vor anderen Studierenden und dem Veranstalter mit anschließender Diskussion. Eine schriftliche Zusammenfassung wird gefordert. Das Proseminar dient auch dazu, die Kommunikationsfähigkeit der Studierenden in einem fachlichen Kontext zu trainieren. Für die Bewertung werden also das erzielte Verständnis des Stoffes, die Selbständigkeit der Vorbereitung, die schriftliche Zusammenfassung und besonders die Qualität des Vortrages in fachlicher und in gestalterischer Hinsicht herangezogen.

### Voraussetzungen für die Teilnahme

### Detailangaben zum Abschluss

Fachgebietsindividuelle Prüfungsleistung

## Proseminar für IN Bsc

Fachabschluss: Studienleistung alternativ Art der Notengebung: Testat / Generierte  
 Sprache: Deutsch Pflichtkennz.: Pflichtfach Turnus: ganzjährig

Fachnummer: 100956 Prüfungsnummer: 2200410

Fachverantwortlich: Prof. Dr. Winfried Kühnhauser

Leistungspunkte: 2 Workload (h): 60 Anteil Selbststudium (h): 38 SWS: 2.0  
 Fakultät für Informatik und Automatisierung Fachgebiet: 2255

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			
													0	2	0																					

### Lernergebnisse / Kompetenzen

Das Proseminar dient dem Einüben des Umgangs mit wissenschaftlichen und/oder anspruchsvollen technischen Texten in rezipierender und darstellender Weise. Hauptaufgabe eines Studierenden ist also die eigenständige, aber betreute Erarbeitung eines Stückes oder mehrerer Stücke fremder wissenschaftlicher Literatur bis zum eigenen Verständnis und die geschlossene Darstellung dieses Materials in einem Vortrag vor anderen Studierenden und dem Veranstalter, mit Befragung und Diskussion auf wissenschaftlichem Niveau (Fachkompetenz, Methodenkompetenz). Eine schriftliche Zusammenfassung wird gefordert. Das Hauptseminar dient auch dazu, die Kommunikationsfähigkeit der Studierenden in einem fachlichen Kontext zu trainieren und zu bewerten (Sozialkompetenz). Für die Bewertung sind also der erzielte Grad von Verständnis des Stoffes, die Selbstständigkeit der Vorbereitung und besonders die Qualität des Vortrages in fachlicher und in gestalterischer Hinsicht heranzuziehen.

Vorkenntnisse  
 fachspezifisch

Inhalt  
 Wechselndes Angebot aus den Informatik-Fachgebieten

Medienformen  
 fachspezifisch

Literatur  
 fachspezifisch

Detailangaben zum Abschluss  
 keine

verwendet in folgenden Studiengängen:  
 Bachelor Informatik 2013  
 Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2013 Vertiefung  
 Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2013 Vertiefung

---

## Modul: Computergrafik

Modulnummer: 100680

Modulverantwortlich: Prof. Dr. Beat Brüderlin

Modulabschluss: Fachprüfung/Modulprüfung generiert

Lernergebnisse

siehe Fachbeschreibung

Voraussetzungen für die Teilnahme

Detailangaben zum Abschluss

schriftlich 60. min.

## Computergrafik

Fachabschluss: Prüfungsleistung schriftlich 60 min Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch Pflichtkennz.: Pflichtfach Turnus: Wintersemester

Fachnummer: 5367

Prüfungsnummer: 2200060

Fachverantwortlich: Prof. Dr. Beat Brüderlin

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 105	SWS: 4.0							
Fakultät für Informatik und Automatisierung			Fachgebiet: 2252							
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS
Fach-	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P
semester					3 1 0					

### Lernergebnisse / Kompetenzen

Vermitteln der Grundlagen der Computergrafik bestehend aus Lineare Algebra/homogene Vektorräumen, Physik des Lichts, Rasteroperationen, Bildsynthese, Bildverarbeitung und effiziente geometrische Algorithmen und Datenstrukturen. Die Vorlesung bildet die Grundlagen für "photorealistische" Bildsynthese, wie sie in der Industrie sowie bei den Medien Verwendung finden (z. B. Filmindustrie, Computer-Aided Design, Computerspiele, Styling). Vermittlung von Grundlagen für weiterführende Vorlesungen: Geometrisches Modellieren, Interaktive Grafische Systeme / Virtuelle Realität, Technisch-wissenschaftliche Visualisierung, Fortgeschrittene Bildsynthese, Bildverarbeitung I & II.

### Vorkenntnisse

Programmierkenntnisse Grundlagen Algorithmen & Datenstrukturen

### Inhalt

Einführung: Überblick über das Fach Grafische Datenverarbeitung. Einführung: Vektoren und Matrizen, Transformationen, Homogene Vektorräume, 2D, 3D-Primitiven und Operationen, View-Transformationen Farbwahrnehmung, Tristimulus Ansatz, Farbmodelle: RGB, CMY, HSV, CIE. Spektrale Ansätze. Additive und Subtraktive Mischung. Lichtquellen und Filter. Rastergrafik-Hardware: Farbdiskretisierung, Farbbildröhre, LCD, Laserprinter, Ink-jet, etc. Rastergrafik: Rasterkonvertierung von Linien und Polygonen (Bresenham-Algorithmus, Polygonfüll-Algorithmus). Bildbearbeitung und Erkennung: Operationen auf dem Bildraster, Bildtransformationen (Skalierung, Drehung), Resampling und Filterung (Bilinear, Gauß) Dithering, Antialiasing, Flood Filling, Kantenverstärkung (Kantenerkennung) Licht und Beleuchtung: (physikalische Größen: Wellenlänge, Leuchtdichte, Leuchtstärke), Wechselwirkung von Licht und Material, Lichtausbreitung und Reflexion, Refraktion, Beleuchtungsmodelle, Materialeigenschaften (geometrische Verteilung) Farbige Lichtquellen (spektrale Verteilung) (Phong: diffuse, spekulare Reflexion). Cook-Torrance, Mehrfachreflexion, Lichteffekte: Schatten, Halbschatten, Kaustik. Bildsynthese: Rendering basierend auf Rasterkonvertierung: Z-Buffer, Flat-Shading, Gouraud shading, Phong Shading Global Illumination, Raytracing, Photontracing, Radiosity Texturemapping / Image-based Rendering: Affines und perspektivisches Texturemapping, projektives Texturemapping, Environment Mapping, Bumpmaps Effiziente Datenstrukturen zum räumlichen Sortieren und Suchen. Kd-Tree, Hüllkörper-Hierarchie, Anwendungen in der Grafik Ray-tracing, Kollisionserkennung. OpenGL, GPU-Renderpipeline, Szenegraphen, Effizientes Rendering grosser Szenen. Ausblick: Überblick geometrischer und physikalischer Modelldatenstrukturen: CSG, B-Rep, Voxel, Octree, parametrische Flächen Computergrafische Animation: (Key frame, motion curve, physikalisch basiertes Modellieren, Kollisionserkennung, Molekülmodelle)

### Medienformen

Tafel, Folien, Buch Brüderlin, Meier: Computergrafik und geometrisches Modellieren (s. unten)

### Literatur

Brüderlin, B., Meier, A., Computergrafik und geometrisches Modellieren, Teubner-Verlag, 2001 Weiterführende Literatur: José Encarnação, Wolfgang Straßer, Reinhard Klein: Graphische Datenverarbeitung 1: Gerätetechnik, Programmierung und Anwendung graphischer Systeme. 4th, revised and extended edition, Oldenbourg, Munich, Germany, 1996. José Encarnação, Wolfgang Straßer, Reinhard Klein: Graphische Datenverarbeitung 2: Modellierung komplexer Objekte und photorealistische Bilderzeugung. 4th, revised and extended edition, Oldenbourg, Munich, Germany, 1997. James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes: Computer Graphics: Principles and Practice, Second Edition in C. - 2nd edition, Addison-Wesley, Reading, MA, USA, 1990. Alan Watt: 3D-Computergrafik. 3rd edition, Addison-Wesley, Reading, MA, USA, 2001.

## Detailangaben zum Abschluss

schriftlich 60. min.

## verwendet in folgenden Studiengängen:

Bachelor Informatik 2010

Bachelor Informatik 2013

Bachelor Ingenieurinformatik 2008

Bachelor Ingenieurinformatik 2013

Bachelor Mathematik 2009

Bachelor Mathematik 2013

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2008 Vertiefung

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2013 Vertiefung

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2008 Vertiefung

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2013 Vertiefung

Master Mathematik und Wirtschaftsmathematik 2008

Master Mathematik und Wirtschaftsmathematik 2013 Vertiefung AM

Master Mathematik und Wirtschaftsmathematik 2013 Vertiefung WM

## Modul: Logik und Logikprogrammierung

Modulnummer: 100339

Modulverantwortlich: Prof. Dr. Dietrich Kuske

Modulabschluss: Fachprüfung/Modulprüfung generiert

### Lernergebnisse

Die Studierenden können zentrale Begriffe der Aussagen- und der Prädikatenlogik darstellen und auf konkrete Problemstellungen anwenden. Sie verstehen das Zusammenspiel von Syntax und Semantik formaler Systeme und argumentieren dabei präzise und folgerichtig. Sie kennen klassische Entscheidungsverfahren für die genannten Kalküle und können sie hinsichtlich ihrer Anwendbarkeit beurteilen. Sie haben ein grundlegendes Verständnis für die Logik-Programmierung und sind in der Lage, typische Probleme durch Prolog-Programme zu lösen.

### Voraussetzungen für die Teilnahme

### Detailangaben zum Abschluss

## Logik und Logikprogrammierung

Fachabschluss: Prüfungsleistung schriftlich 150 min Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch Pflichtkennz.: Pflichtfach Turnus: Sommersemester

Fachnummer: 100574 Prüfungsnummer: 2200381

Fachverantwortlich: Prof. Dr. Dietrich Kuske

Leistungspunkte: 5 Workload (h): 150 Anteil Selbststudium (h): 94 SWS: 5.0  
 Fakultät für Informatik und Automatisierung Fachgebiet: 2241

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS					
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P
																3	2	0															

### Lernergebnisse / Kompetenzen

Die Studierenden können zentrale Begriffe der Aussagen- und der Prädikatenlogik darstellen und auf konkrete Problemstellungen anwenden. Sie verstehen das Zusammenspiel von Syntax und Semantik formaler Systeme und argumentieren dabei präzise und folgerichtig. Sie kennen klassische Entscheidungsverfahren für die genannten Kalküle und können sie hinsichtlich ihrer Anwendbarkeit beurteilen. Sie haben ein grundlegendes Verständnis für die Logik-Programmierung und sind in der Lage, typische Probleme durch Prolog-Programme zu lösen.

### Vorkenntnisse

sicherer Umgang mit mengentheoretischen Begriffen und Notationen (z.B. erworben in "Grundlagen und Diskrete Strukturen")

### Inhalt

Aussagenlogik: Syntax und Semantik, Erfüllbarkeit, Allgemeingültigkeit, Folgerung, Normalformen, Wahrheitswerttabellen, Resolution, natürliches Schließen, Tableau-Methode

Prädikatenlogik: Syntax und Semantik, Erfüllbarkeit, Allgemeingültigkeit, Folgerung, Normalformen, Resolution, Kompaktheitssatz, natürliches Schließen, Unentscheidbarkeit der Allgemeingültigkeit von Formeln der Prädikatenlogik und der Theorie der natürlichen Zahlen

PROLOG-Programmierung an typischen Problemklassen

### Medienformen

Folien, Übungsblätter, Vorlesungsmitschnitte (online)

### Literatur

M. Huth, M. Ryan: Logic in Computer Science, Cambridge 2010  
 S. Hölldobler: Logik und Logikprogrammierung, Krottenmühl 2009  
 Ivan Bratko: Prolog Programming for Artificial Intelligence. 4th edition, ISBN-10: 0321417461, ISBN-13: 9780321417466, Addison-Wesley, 2012.

### Detailangaben zum Abschluss

### verwendet in folgenden Studiengängen:

Bachelor Informatik 2013  
 Bachelor Ingenieurinformatik 2013  
 Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2013 Vertiefung  
 Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2013 Vertiefung



---

## Modul: Wahlpflichtmodul(Wahl eines Moduls)

Modulnummer: 101210

Modulverantwortlich: Prof. Daniel Ziener

Modulabschluss: Fachprüfung/Modulprüfung generiert

### Lernergebnisse

Die Modulbeschreibung ergibt sich aus den Beschreibungen der Module, die in diesem Modul enthalten sind.

### Voraussetzungen für die Teilnahme

Die Modulbeschreibung ergibt sich aus den Beschreibungen der Module, die in diesem Modul enthalten sind.

### Detailangaben zum Abschluss

## Modul: Rechnerarchitekturen für IN

Modulnummer: 100331

Modulverantwortlich: Prof. Daniel Ziener

Modulabschluss: Fachprüfung/Modulprüfung generiert

### Lernergebnisse

Allgemein:

Umfassende Kenntnisse über Grundlagen und fortgeschrittene Konzepte von Rechnerarchitekturen, praktische Erfahrungen aus Übungsbeispielen und Laborversuchen

Detailliert:

Siehe Einzelfächer

### Vorraussetzungen für die Teilnahme

### Detailangaben zum Abschluss

- Schriftliche Prüfungen "Rechnerarchitekturen 1" und "Rechnerarchitekturen 2"
- Testat aus dem Praktikum
- Notenbildung erfolgt aus den Prüfungen

# Rechnerarchitekturen 1

Fachabschluss: Prüfungsleistung schriftlich 90 min Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch Pflichtkennz.: Pflichtfach Turnus: Sommersemester

Fachnummer: 5382 Prüfungsnummer: 2200265

Fachverantwortlich: Prof. Daniel Ziener

Leistungspunkte: 4 Workload (h): 120 Anteil Selbststudium (h): 75 SWS: 4.0  
 Fakultät für Informatik und Automatisierung Fachgebiet: 2231

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			
										2	2	0																								

## Lernergebnisse / Kompetenzen

**Fachkompetenz:**  
 Die Studierenden verstehen detailliert Aufbau und Funktionsweise von Prozessoren, typischen Rechnerbaugruppen und deren Zusammenwirken.

**Methodenkompetenz:**  
 Die Studierenden sind in der Lage, ein Beschreibungsmittel für die Modellierung von Strukturen und Abläufen mit formalen Mitteln anzuwenden. Die Studierenden entwerfen und analysieren einfache maschinennahe Programme. Die Studierenden konzipieren und entwerfen einfache Speicher- und E/A-Baugruppen.

**Systemkompetenz:**  
 Die Studierenden verstehen das Zusammenwirken der Funktionsgruppen von Rechnern als System und in Rechnersystemen. Sie erkennen den Zusammenhang zwischen Architektur und Anwendung auf dem Maschinenniveau anhand praktischer Übungen.

**Sozialkompetenz:**  
 Die Studierenden sind in der Lage, Problemstellungen der Rechnerarchitektur in der Gruppe zu lösen.

## Vorkenntnisse

Vorlesung und Übung „Rechnerorganisation“

## Inhalt

- Begriff der Rechnerarchitektur,
- Architekturmodellierung mit Systemen gekoppelter Automaten,
- Innenarchitektur von Prozessoren,
- Befehlssatzarchitektur und Assemblerprogramme,
- Außenarchitektur von Prozessoren,
- Aufbau und Funktion von Speicherbaugruppen
- Aufbau und Funktion von Ein- und Ausgabebaugruppen,
- Zusammenwirken von Rechnerbaugruppen im Gesamtsystem

## Medienformen

Vorlesung: Folien, Arbeitsblätter (Online und Copyshop)  
 Übung: Arbeitsblätter und Aufgabensammlung (Online und Copyshop)  
 Selbststudium: Teleteaching-Kurs  
 Allgemein: Webauftritt (Materialsammlung, Teleteaching-Kurs, Literaturhinweise, Links und weiterführende Infos)

## Literatur

**Primär:**

- W. Fengler und O. Fengler: Grundlagen der Rechnerarchitektur. Ilmenau 2016. ilmedia.

Weiteres eigenes Material: Materialsammlung zum Download - Materialsammlung im Copyshop - Teleteaching-Kurs

**Sekundär:**

- W. Fengler, I. Philippow: Entwurf Industrieller Mikrocomputer-Systeme. ISBN 3-446-16150-3, Hanser 1991.
- C. Martin: Einführung in die Rechnerarchitektur - Prozessoren und Systeme. ISBN 3-446-22242-1, Hanser 2003.

- T. Flik: Mikroprozessortechnik und Rechnerstrukturen. ISBN 3-540-22270-7, Springer 2005.  
Allgemein: Webauftritt! <http://tu-ilmenau.de/?r-ra1>

Detailangaben zum Abschluss

verwendet in folgenden Studiengängen:

Bachelor Informatik 2010

Bachelor Informatik 2013

Bachelor Mathematik 2009

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2013 Vertiefung

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2013 Vertiefung

## Rechnerarchitekturen 2

Fachabschluss: Prüfungsleistung schriftlich 90 min Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch Pflichtkennz.: Pflichtfach Turnus: Wintersemester

Fachnummer: 5383 Prüfungsnummer: 2200055

Fachverantwortlich: Prof. Daniel Ziener

Leistungspunkte: 3 Workload (h): 90 Anteil Selbststudium (h): 56 SWS: 3.0  
 Fakultät für Informatik und Automatisierung Fachgebiet: 2231

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			
													2	1	0																					

### Lernergebnisse / Kompetenzen

**Fachkompetenz:**  
 Die Studierenden verstehen detailliert Aufbau und Funktionsweise von fortgeschrittenen Prozessoren und Rechnern. Die Studierenden verstehen Entwicklungstendenzen der modernen Rechner- und Systemarchitektur.

**Methodenkompetenz:**  
 Die Studierenden sind in der Lage, Anwendungsbeispiele und Architekturvarianten zu entwickeln. Die Studierenden analysieren Leistungskennwerte von Rechnern und Rechnersystemen.

**Systemkompetenz:**  
 Die Studierenden verstehen das Zusammenwirken der Funktionsgruppen von fortgeschrittenen Rechnern als System und in Rechnersystemen. Sie erkennen den Zusammenhang zwischen Architektur, Leistung und Anwendung anhand von Übungsbeispielen.

**Sozialkompetenz:**  
 Die Studierenden sind in der Lage, Problemstellungen der Rechnerarchitektur in der Gruppe zu lösen.

### Vorkenntnisse

Vorlesung und Übung „Rechnerarchitekturen 1“ oder vergleichbare Veranstaltung

### Inhalt

- Entwicklung der Prozessorarchitektur: Complex-Instruction-Set-Computing (CISC), Reduced-Instruction-Set-Computing (RISC); #
- Befehls-Pipelining;
- Skalare Prozessorarchitektur, Very-Long-Instruction-Word-Architektur, Out of Order-Execution; Simultaneous Multithreading.
- Entwicklung der Speicherarchitektur: Adresspipelining, Burst Mode und Speicher-Banking;
- Speicherhierarchie,
- Cache-Prinzip, Cache-Varianten;
- Beispielarchitekturen;
- Spezialrechner:
- Aufbau eines Einchip-Controllers; Einchipmikrorechner des mittleren Leistungssegments, Erweiterungen im E/A-Bereich;
- Prinzip der digitalen Signalverarbeitung, Digitale Signalprozessoren (DSP), Spezielles Programmiermodell;
- Leistungsbewertung: MIPS, MFLOPS; Speicherbandbreite; Programmabhängiges Leistungsmodell (Benchmarkprogramme);
- Parallele Rechnerarchitekturen: Single Instruction Multiple Data, Multiple Instruction Single Data, Multiple Instruction Multiple Data; Enge und Lose Kopplung, Verbindungstopologien
- Entwicklung von Anwendungsbeispielen, Architekturvarianten und Berechnung von Leistungskennwerten

### Medienformen

Vorlesung: Folien, Arbeitsblätter (Online und Copyshop)  
 Übung: Arbeitsblätter und Aufgabensammlung (Online und Copyshop)  
 Allgemein: Webaufttritt (Materialsammlung, Literaturhinweise, Links und weiterführende Infos)

### Literatur

Primär: Eigenes Material (Online und Copyshop)  
 Sekundär:

- C. Märtin: Einführung in die Rechnerarchitektur - Prozessoren und Systeme. ISBN 3-446-22242-1, Hanser 2003.
  - J. L. Hennessy, D. A. Patterson: Rechnerorganisation und -entwurf. ISBN 3-8274-1595-0, Elsevier 2005.
  - W. Stallings: Computer Organization & Architecture. ISBN 0-13-035119-9, Prentice Hall 2003.
  - A. S. Tanenbaum, J. Goodman: Computerarchitektur. ISBN 3-8273-7016-7, Pearson Studium 2003.
- Allgemein: Webaufttritt: <http://tu-ilmenau.de/?r-ra2>

Detailangaben zum Abschluss

verwendet in folgenden Studiengängen:

Bachelor Informatik 2010

Bachelor Informatik 2013

Bachelor Ingenieurinformatik 2008

Bachelor Ingenieurinformatik 2013

Bachelor Mathematik 2009

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2013 Vertiefung

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2013 Vertiefung

## Praktikum Rechnerarchitekturen 1 und 2

Fachabschluss: Studienleistung alternativ Art der Notengebung: Testat / Generierte  
 Sprache:deutsch Pflichtkennz.:Pflichtfach Turnus:ganzjährig

Fachnummer: 100561 Prüfungsnummer:2200374

Fachverantwortlich: Dr. Bernd Däne

Leistungspunkte: 1 Workload (h):30 Anteil Selbststudium (h):19 SWS:1.0  
 Fakultät für Informatik und Automatisierung Fachgebiet:2231

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS					
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P
																0	0	1															

### Lernergebnisse / Kompetenzen

**Fachkompetenz:**  
 Die Studierenden verfügen über Kenntnisse und Überblickswissen zur Funktionsweise von Prozessoren und Rechnerstrukturen. Sie beherrschen den Umgang mit Beschreibungsmitteln und Modellen und erkennen das Zusammenwirken von Hardware und Software auf hardwarenahen Architekturebenen.

**Methodenkompetenz:**  
 Die Studierenden sind in der Lage, maschinennahe Programme zu verstehen, zu erstellen und in Betrieb zu nehmen. Sie sind in der Lage zur werkzeuggestützten Modellierung und zur Simulation und Analyse von Modellen.

**Systemkompetenz:**  
 Sie beherrschen den Umgang mit Werkzeugen zu Programmerstellung, Programmtest, Modellerstellung und Modellanalyse.

**Sozialkompetenz:**  
 Die Studierenden erarbeiten Problemlösungen gemeinsam in kleinen Gruppen.

### Vorkenntnisse

Vorlesung und Übung Rechnerarchitekturen 1 (oder vergleichbare Veranstaltung)  
 Vorlesung und Übung Rechnerarchitekturen 2 (oder vergleichbare Veranstaltung)

### Inhalt

Einfache Assemblerprogramme  
 Ein- und Ausgabebaugruppen  
 Microcontroller  
 Fortgeschrittene Pipeline-Architekturen

### Medienformen

Laborpraktikum. Gedruckte Anleitungen, Ergänzende Onlinequellen, Hilfefunktionen der benutzten Software.

### Literatur

Unterrichtsmaterialien und empfohlene Literatur der Fächer Rechnerarchitekturen 1 und Rechnerarchitekturen 2. Zusätzliche Literaturhinweise und weitere Quellen befinden sich stets auf den Internetseiten zur Lehrveranstaltung:  
<http://tu-ilmenau.de/?r-p-ra1>  
<http://tu-ilmenau.de/?r-p-ra2>

### Detailangaben zum Abschluss

Es sind zwei Versuche zu RA 1 und zwei Versuche zu RA 2 durchzuführen und mittels Testat nachzuweisen.

### verwendet in folgenden Studiengängen:

Bachelor Informatik 2013  
 Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2013 Vertiefung  
 Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2013 Vertiefung

## Modul: Neuroinformatik und Schaltsysteme

Modulnummer: 100332

Modulverantwortlich: Prof. Dr. Horst-Michael Groß

Modulabschluss: Fachprüfung/Modulprüfung generiert

### Lernergebnisse

**Fachkompetenz:** Die Studierenden lernen in diesem Modul die wesentlichen Grundlagen der sequentiellen und der parallelen (konnektionistischen) Informationsverarbeitung als die zwei wesentlichen Paradigmen der Informatik kennen.

Im Teil Neuroinformatik werden die Grundlagen der parallelen neurobiologischen Informationsverarbeitung und der darauf aufbauenden Neuroinformatik als wesentliche Säule der "Computational Intelligence" vermittelt. Die Studierenden verstehen die grundsätzliche Herangehensweise des konnektionistischen Ansatzes und kennen die wesentlichen biologischen Grundlagen, mathematischen Modellierungs- und algorithmischen Implementierungstechniken beim Einsatz von neuronalen und probabilistischen Methoden im Unterschied zu klassischen Methoden der Informations- und Wissensverarbeitung.

Im Teil Schaltsysteme verfügen die Studierenden über Kenntnisse und vertieftes Wissen zu speziellen Strukturen und Funktionen von digitaler und programmierbarer Hardware und haben ein vertieftes Verständnis für die praktisch relevanten Problemstellungen und deren Komplexität.

**Methodenkompetenz:** Die Studierenden sind in der Lage, Fragestellungen aus den o. g. Problemkreisen zu analysieren, durch Anwendung des behandelten Methodenspektrums neue Lösungskonzepte zu entwerfen und umzusetzen, sowie bestehende Lösungen zu bewerten. Die Studierenden können ausgehend von einer Problemanalyse eigene Lösungen mit neuronalen Techniken erstellen. Sie sind in der Lage, komplexe digitale Schaltungen zu analysieren und zu synthetisieren und können auch umfangreichere Steuerungen sowohl mit Hilfe von diskreten Gatterschaltungen als auch mit Hilfe programmierbarer Schaltkreise erstellen. Sie können beim Entwurf systematisch vorgehen und ihre Entwürfe verifizieren.

**Systemkompetenz:** Auf Basis der vermittelten Methodik sind die Studierenden in der Lage, Methoden der Computational Intelligence auf neue Probleme anzuwenden und erfolgreich einzusetzen. Sie können dabei auf ein breites Methodenwissen aus den Bereichen der Neuroinformatik zurückgreifen. Im Teil Schaltsysteme sind die Studierenden in der Lage, Programmsysteme zum Entwurf digitaler Steuerungen und Schaltungen anzuwenden.

**Sozialkompetenz:** Die Studierenden sind in der Lage, praktische Problemstellungen mit Methoden der parallelen und sequentiellen Informationsverarbeitung in der Gruppe zu analysieren, zu lösen und die Lösungen zu präsentieren.

Sie erarbeiten Problemlösungen komplexer digitaler Schaltungen in der Gruppe, wobei einzelne Teilfunktionen von unterschiedlichen Personen entworfen werden. Sie können die von ihnen synthetisierten Schaltungen und Modellsteuerungen gemeinsam in einem Praktikum erproben, auf Fehler analysieren und korrigieren

### Voraussetzungen für die Teilnahme

### Detailangaben zum Abschluss



## Neuroinformatik

Fachabschluss: über Komplexprüfung

Art der Notengebung: unbenotet

Sprache: Deutsch

Pflichtkennz.: Pflichtfach

Turnus: Sommersemester

Fachnummer: 1389

Prüfungsnummer: 2200343

Fachverantwortlich: Prof. Dr. Horst-Michael Groß

Leistungspunkte: 3	Workload (h): 90	Anteil Selbststudium (h): 56	SWS: 3.0							
Fakultät für Informatik und Automatisierung			Fachgebiet: 2233							
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS
Fach-	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P
semester				2 1 0						

### Lernergebnisse / Kompetenzen

In der Vorlesung "Neuroinformatik und Maschinelles Lernen" lernen die Studierenden die konzeptionellen, methodischen und algorithmischen Grundlagen der Neuroinformatik und des Maschinellen Lernens kennen. Ein Sie verstehen die grundsätzliche Herangehensweise dieser Form des Wissenserwerbs, der Generierung von Wissen aus Beobachtungen und Erfahrungen. Sie verstehen, wie ein künstliches System aus Trainingsbeispielen lernt und diese nach Beendigung der Lernphase verallgemeinern kann. Dabei werden die Beispiele nicht einfach auswendig gelernt, sondern das System „erkennt“ Muster und Gesetzmäßigkeiten in den Lerndaten. Die Studierenden lernen die wesentlichen Konzepte, Lösungsansätze sowie Modellierungs- und Implementierungstechniken beim Einsatz von neuronalen und probabilistischen Methoden der Informations- und Wissensverarbeitung kennen. Die Studierenden sind in der Lage, praxisorientierte Fragestellungen aus dem o. g. Problemkreis zu analysieren, durch Anwendung des behandelten Methodenspektrums auf Fragestellungen aus den behandelten Bereichen (Signal-, Sprach- und Bildverarbeitung, Robotik und autonome Systeme, Assistenzsysteme, Mensch-Maschine Interaktion) neue Lösungskonzepte zu entwerfen und algorithmisch (Fokus auf Python) umzusetzen sowie bestehende Lösungen zu bewerten.

### Vorkenntnisse

Keine

### Inhalt

Die Lehrveranstaltung vermittelt das erforderliche Methodenspektrum aus theoretischen Grundkenntnissen und praktischen Fähigkeiten zum Verständnis, zur Implementierung und zur Anwendung neuronaler und probabilistischer Techniken des Wissenserwerbs durch Lernen aus Erfahrungsbeispielen sowie zur Informations- und Wissensverarbeitung in massiv parallelen Systemen. Sie vermittelt sowohl Faktenwissen, begriffliches und algorithmisches Wissen aus folgenden Themenkomplexen:

Intro: Begriffsbestimmung, Literatur, Lernparadigmen (Unsupervised / Reinforcement / Supervised Learning), Haupteinsatzgebiete (Klassifikation, Clusterung, Regression, Ranking), Historie Neuronale Basisoperationen und Grundstrukturen:

- Neuronenmodelle: Biologisches Neuron, I&F Neuron, Formale Neuronen
- Netzwerkmodelle: Grundlegende Verschaltungsprinzipien & Architekturen

Lernparadigmen und deren klassische Vertreter:

- Unsupervised Learning: Vektorquantisierung, Self-Organizing Feature Maps, Neural Gas, k-Means
- Clustering
  - Reinforcement Learning: Grundbegriffe, Q-Learning
  - Supervised Learning: Perzeptron, Multi-Layer-Perzeptron & Error-Backpropagation-Lernregel, RBF-Netze, Expectation-Maximization Algorithmus, Support Vector Machines (SVM)
- Moderne Verfahren für große Datensets

- Deep Neural Networks: Grundidee, Arten, Convolutional Neural Nets (CNN)

Anwendungsbeispiele: Signal-, Sprach- und Bildverarbeitung, Robotik und autonome Systeme, Assistenzsysteme, Mensch-Maschine Interaktion

Exemplarische Software-Implementationen neuronaler Netze für unüberwachte und überwachte Lern- und Klassifikationsprobleme (Fokus auf Python)

Die Studierenden erwerben somit auch verfahrensorientiertes Wissen, indem für reale Klassifikations- und

Lernprobleme verschiedene neuronale Lösungsansätze theoretisch behandelt und praktisch umgesetzt werden. Im Rahmen des Pflichtpraktikums werden die behandelten methodischen und algorithmischen Grundlagen der neuronalen und probabilistischen Informationsverarbeitungs- und Lernprozesse durch die Studierenden mittels interaktiver Demo-Applets vertieft und in Gesprächsgruppen aufgearbeitet.

#### Medienformen

Powerpoint-Folien (als Papierkopie oder PDF), Demo-Apps, Videos, Python Demo Code

#### Literatur

- Zell, A.: Simulation Neuronaler Netzwerke, Addison-Wesley 1997
- Bishop, Ch.: Pattern Recognition and Machine Learning, Springer 2006
- Alpaydin, Ethem: Maschinelles Lernen, Oldenbourg Verlag 2008
- Murphy, K. : Machine Learning – A Probabilistic Perspective, MIT Press 2012
- Goodfellow, I. et al.: Deep Learning, MIT Press 2016

#### Detailangaben zum Abschluss

#### verwendet in folgenden Studiengängen:

Bachelor Biomedizinische Technik 2008  
Bachelor Elektrotechnik und Informationstechnik 2008  
Bachelor Informatik 2010  
Bachelor Informatik 2013  
Bachelor Ingenieurinformatik 2008  
Bachelor Ingenieurinformatik 2013  
Bachelor Mathematik 2009  
Bachelor Mathematik 2013  
Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2013 Vertiefung  
Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2013 Vertiefung  
Master Mathematik und Wirtschaftsmathematik 2008  
Master Wirtschaftsinformatik 2013  
Master Wirtschaftsinformatik 2014  
Master Wirtschaftsinformatik 2015  
Master Wirtschaftsinformatik 2018

## Schaltsysteme

Fachabschluss: über Komplexprüfung

Art der Notengebung: unbenotet

Sprache: Deutsch

Pflichtkennz.: Pflichtfach

Turnus: Sommersemester

Fachnummer: 100457

Prüfungsnummer: 2200344

Fachverantwortlich: Dr. Heinz-Dietrich Wuttke

Leistungspunkte: 3	Workload (h): 90	Anteil Selbststudium (h): 56	SWS: 3.0																											
Fakultät für Informatik und Automatisierung		Fachgebiet: 2235																												
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS																				
Fach-	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P
semester							2	1	0																					

### Lernergebnisse / Kompetenzen

#### Lernziele:

- vertiefende Vermittlung von fundierten Kenntnissen und Fertigkeiten zum Entwurf digitaler Systeme,
- Einbeziehung verallgemeinerter Wertverlaufsgleichheiten,
- Herausbildung von Fähigkeiten zur kritischen Beurteilung von entworfenen Schaltsystemen bzgl. Aufwand und Korrektheit sowie zur praktischen Fehlersuche in Hard- und Softwarerealisierungen

#### Fachkompetenz:

Die Studierenden verfügen über Kenntnisse und vertieftes Wissen zu speziellen Strukturen und Funktionen von digitaler und programmierbarer Hardware und haben ein vertieftes Verständnis für die praktisch relevanten Problemstellungen und deren Komplexität.

#### Methodenkompetenz:

Die Studierenden sind in der Lage, komplexe digitale Schaltungen zu analysieren und zu synthetisieren. Sie können auch umfangreichere Steuerungen sowohl mit Hilfe von diskreten Gatterschaltungen als auch mit Hilfe programmierbarer Schaltkreise erstellen. Sie können beim Entwurf systematisch vorgehen und ihre Entwürfe verifizieren.

#### Systemkompetenz:

Die Studierenden sind in der Lage, Programmsysteme zum Entwurf digitaler Steuerungen und Schaltungen anzuwenden.

#### Sozialkompetenz:

Die Studierenden erarbeiten Problemlösungen komplexer digitaler Schaltungen in der Gruppe, wobei einzelne Teilfunktionen von unterschiedlichen Personen entworfen werden. Sie können die von ihnen synthetisierten Schaltungen und Modellsteuerungen gemeinsam in einem Praktikum erproben, auf Fehler analysieren und korrigieren

### Vorkenntnisse

- erfolgreicher Abschluß des Fachs "Rechnerorganisation"
- Grundkenntnisse im Entwurf kombinatorischer und sequentieller Schaltungen

### Inhalt

Einführung

Entwurf kombinatorischer Schaltungen

- Verallgemeinerte Wertverlaufsgleichheit
- Implizite Gleichungssysteme
- Struktursynthese, Minimierung
- Dynamische Probleme

Entwurf sequentieller Automaten

- Partielle, nichtdeterminierte Automaten
- Struktursynthese mit unterschiedlichen Flip-Flop-Typen
- Operations- und Steuerwerke

#### Entwurf paralleler Automaten

- Komposition/ Dekomposition
- Automatenetze

#### Entwurfswerkzeuge

#### Medienformen

Vorlesung mit Tafel und PowerPoint, Video zur Vorlesung, Applets und PowerPoint-Präsentationen im Internet, Arbeitsblätter, Lehrbuch

#### Literatur

- Wuttke, Henke: Schaltsysteme, Pearson-Verlag, München 2003
- -Informatik-Duden: Duden-Verlag 1988/89Schiffmann,
- S. Hentschke: Grundzüge der Digitaltechnik, Teubner-Verlag, Stuttgart 1988
- T. Flick, H. Liebig: Mikroprozessortechnik, 4. Auflage, Springer- Verlag, Berlin 1994
- Literaturempfehlungen zu den Vorlesungen

#### Detailangaben zum Abschluss

Die 20-minütige mündliche Prüfung geht zu 50% in die Modulprüfung Neuroinformatik/Schaltsysteme ein.

#### verwendet in folgenden Studiengängen:

Bachelor Informatik 2013

Bachelor Ingenieurinformatik 2013

Bachelor Mathematik 2013

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2013 Vertiefung

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2013 Vertiefung

Master Wirtschaftsinformatik 2013

Master Wirtschaftsinformatik 2014

Master Wirtschaftsinformatik 2015

Master Wirtschaftsinformatik 2018

## Praktikum Neuroinformatik und Schaltsysteme

Fachabschluss: Studienleistung Art der Notengebung: Testat / Generierte  
 Sprache: Deutsch Pflichtkennz.: Pflichtfach Turnus: Sommersemester

Fachnummer: 100536

Prüfungsnummer: 2200372

Fachverantwortlich: Dr. Klaus Debes

Leistungspunkte: 2	Workload (h): 60	Anteil Selbststudium (h): 38	SWS: 2.0							
Fakultät für Informatik und Automatisierung			Fachgebiet: 2233							
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS
Fach-	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P
semester					0 0 2					

### Lernergebnisse / Kompetenzen

#### Fachkompetenz:

SS: Durch die vertiefende Vermittlung von fundierten Kenntnissen und Fertigkeiten zum Entwurf digitaler, sequentieller Steuerungssysteme sowie Möglichkeiten zu deren formaler Beschreibung und Verifikation.

NI: Durch die Vermittlung von fundierten Grundlagen zur Funktionsweise von überwachten und unüberwachten neuronalen Lernverfahren.

#### Methodenkompetenz:

SS: Die Studierenden sind in der Lage, digitale Steuerungen zu analysieren, zu optimieren und zu synthetisieren. Sie können digitale Steuerungen sowohl mit Hilfe von diskreten Gatterschaltungen als auch mit Hilfe programmierbarer Schaltkreise erstellen. Fähigkeiten zur kritischen Beurteilung von entworfenen Schaltsystemen bzgl. Aufwand und Korrektheit befähigen zur praktischen Fehlersuche in Hard- und Softwarerealisierungen.

NI: Die Studierenden sind in der Lage, die Funktionsweise neuronalen Lernverfahren zu verstehen, zu analysieren und eigenständig für unterschiedliche Problemstellungen zu entwerfen.

#### Systemkompetenz:

SS: Mit Hilfe formaler Methoden können sie digitale Steuerungssysteme analysieren und validieren.

NI: Erlernen formaler Methoden zur Anwendung neuronaler überwachter und unüberwachter Lernverfahren;

#### Sozialkompetenz:

SS: Die Studierenden erarbeiten Problemlösungen einfacher digitaler Schaltungen in der Gruppe. Sie können die von ihnen synthetisierten Schaltungen gemeinsam in einem Praktikum auf Fehler analysieren und korrigieren.

NI: Die Studierenden analysieren auf der Basis vorgegebener Applets in kleinen Gruppen und stellen ihre Ergebnisse vor.

### Vorkenntnisse

Vorlesung Neuroinformatik, Vorlesung Schaltsysteme

### Inhalt

NI: Die behandelten methodischen und technischen Grundlagen der neuronalen und probabilistischen Informationsverarbeitungs- und Lernprozesse werden mittels interaktiver Demo-Applets vertieft und in Gesprächsgruppen aufgearbeitet; dazu Erarbeitung eines Protokolls im Selbststudium mit Überprüfung der Ergebnisse in einer Präsenzveranstaltung (Arbeitsaufwand 15 Zeitstunden)

SS: Durchführung von zwei Versuchen

- Hardware-Realisierung sequentieller Schaltungen
- PLD-Realisierung sequentieller Schaltungen

### Medienformen

## Literatur

NI: Praktikumsanleitung und Vorlesungsunterlagen

SS:

- H.D. Wuttke, K. Henke: Schaltsysteme – Eine Automatenorientierte Einführung – Pearson Education, 2006
- H.D. Wuttke, K. Henke: Schaltsysteme, Arbeitsblätter, Übungsaufgaben, Praktikumsanleitung, TU Ilmenau, [www.tu-ilmenau.de/iks](http://www.tu-ilmenau.de/iks)
- W. Schiffmann, H. Schmitz: Technische Informatik, Band I und II, Springer-Verlag, 2004
- V. Claus, A. Schwill: Informatik-Duden, Bibliographisches Institut, 2006
- Spezielle Literatur in den Versuchsanleitungen und unter [www.tu-ilmenau.de/iks](http://www.tu-ilmenau.de/iks)

## Detailangaben zum Abschluss

verwendet in folgenden Studiengängen:

Bachelor Informatik 2013

Bachelor Ingenieurinformatik 2013

Bachelor Mathematik 2013

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2013 Vertiefung

Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2013 Vertiefung



## **Glossar und Abkürzungsverzeichnis:**

LP	Leistungspunkte
SWS	Semesterwochenstunden
FS	Fachsemester
V S P	Angabe verteilt auf Vorlesungen, Seminare, Praktika
N.N.	Nomen nominandum, Platzhalter für eine noch unbekannte Person (wikipedia)
Objekttypen lt. Inhaltsverzeichnis	K=Kompetenzfeld; M=Modul; P,L,U= Fach (Prüfung, Lehrveranstaltung, Unit)