

# **Modulhandbuch Informatik Master of Science**

**Version M\_Inf20.0\_S**

Letzte Änderung: 2024-07-11 08:47:24

# Inhaltsverzeichnis

MM003 – Algorithmics  
MM005 – Funktionale Programmierung  
MM006 – Learning and Softcomputing  
MM009 – Workshop Kryptographie  
MM018 – Robotics  
MM023 – Seminar Informatik  
MM062 – Praktikum Virtuelle Realität und Simulation  
MM162 – Moderne Software-Architekturen  
MM010 – Aktuelle Entwicklungen in der Informatik  
MM027 – Konzepte der Datenbanktechnologie  
MM029 – Berechenbarkeit und Verifikation  
MM033 – Methoden der Künstlichen Intelligenz  
MM035 – Distributed Systems  
MM042 – Digitale Kommunikationssysteme und Reconfigurable Computing  
MM044 – Fotorealismus und Simulation  
MM048 – Projekt Informatik  
MM050 – Master-Thesis  
MM058 – Master-Kolloquium

# Module

## ◆ MM003 - Algorithmics

Verantwortliche:	Sebastian Iwanowski
Moduldauer:	6 Monate
Unterrichtssprache:	english

### Bestandteile:

Teilleistung	Lernform	Prüfungsform	-umfang	-dauer	ECTS	Benotung	Turnus	Aufwand	Lehrende
TM027 - Algorithmics	Vorlesung	Klausur / Mündliche Prüfung			5.0	Drittelnoten	jährlich	150 Stunden	Sebastian Iwanowski

### Lehrinhalte:

- Einführung in die formale Algorithmik
  - Vergleich der grundlegenden Sortiertechniken
  - Komplexitätsmaße für die Analyse von Algorithmen
  - Untere Schranke für Algorithmen, die nur Vergleiche verwenden
- Fortgeschrittenes Suchen und Sortieren
  - Ordnungsstatistik
  - Suche in sortierten Arrays
  - Sortierung in endlichen Bereichen
- Lösungen für das Wörterbuchproblem
  - Hashing und andere Methoden zur Optimierung des durchschnittlichen Laufzeitverhaltens
  - (2,3)-Bäume als Beispiel für einen optimalen Baum für die schlechteste Laufzeit
  - Andere optimale Schlechteste-Fall-Methoden für Suchbäume
  - Optimale binäre Suchbäume (Bellman)
- Graphenalgorithmen
  - Die Erstellung minimal aufspannender Gerüste als Motivation für grundlegende Algorithmen
  - Kürzeste Wege (Dijkstra, Floyd-Warshall, Straßen)
  - Berechnung der maximalen Flüsse in q/s-Netzwerken (Ford-Fulkerson, Edmonds-Karp, Dinic)
  - Berechnung von Graphenmatchings (bipartit, Edmonds)
- String-Matching
- Grundlagen der algorithmischen Geometrie
  - Grundlegende Probleme und die Verwendung von Voronoi-Diagrammen zu ihrer Lösung
  - Sweep-Techniken (einschließlich Berechnung von Voronoi-Diagrammen)

### Qualifikationsziele:

Die Studierenden ...

- kennen die grundlegenden Problemstellungen der Algorithmik und deren klassische Lösungsverfahren.
- können die Korrektheit und Effizienz von Algorithmen analysieren.
- haben detaillierte Kenntnisse über fortgeschrittene Algorithmen für diverse Problemstellungen in ausgewählten Anwendungsbereichen.
- wissen, wie man theoretische Ergebnisse in praktischen Anwendungen implementiert.

### Verwendbarkeit:

Das Modul ist als Einstiegsmodul geeignet. Es legt die theoretischen Grundlagen für ein wissenschaftliches IT-orientiertes Studium. Es umfasst das Wissen über grundlegende Algorithmen, die zur Lösung verschiedener Anwendungsprobleme notwendig sind.

### Voraussetzungen und Empfehlungen:

Diskrete Mathematik aus dem Bachelorstudium, gutes allgemeines Programmierverständnis

### Literatur:

- deBerg, M., Cheong, O., van Krefeld, M., Overmars, M.: Computational Geometry, Algorithms and Applications. Springer 2008 (3. edition), ISBN 978-3540779735
- Cormen, T.; Leiserson C.; Rivest, R.; Stein, C.: Introduction to Algorithms,

MIT Press 2001 (2nd ed.)

- Levitin, A.:  
Introduction to the Design and Analysis of Algorithms.  
Addison-Wesley 2006, ISBN 0-321-36413-9
- Mehlhorn, K. / Sanders, P.:  
Algorithms and Data Structures The Basic Toolbox.  
Springer 2008, ISBN 978-3-540-77977-3
- Papadimitriou, C. / Steiglitz, K.:  
Combinatorial Optimization Algorithms and Complexity.  
Dover 1998, ISBN 0-486-40258-4

### **Studiengänge:**

- Data Science & Artificial Intelligence Master of Science (Wahlmöglichkeit 1. Semester)
- Informatik Master of Science (1. Semester)
- IT Engineering Master of Science (Wahlmöglichkeit 1. Semester)

## ◆ MM005 - Funktionale Programmierung

Verantwortliche:	Torben Tietgen
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

### Bestandteile:

Teilleistung	Lernform	Prüfungsform	-umfang	-dauer	ECTS	Benotung	Turnus	Aufwand	Lehrende
TM028 - Funktionale Programmierung	Vorlesung	Klausur / Mündliche Prüfung		45 Min.	2.0	Drittelnoten	jährlich	60 Stunden	Frank Huch
TM029 - Ubg. Funktionale Programmierung	Übung	Abnahme	6 Aufgaben		3.0	Bestanden/nicht Bestanden	jährlich	90 Stunden	Torben Tietgen

### Lehrinhalte:

- Syntax von Haskell
- Produkt- und Summen-Datentypen
- Rekursive Datenstrukturen
- Polymorphismus
- Funktionen höherer Ordnung
- Hindley-Milner Typsystem, Typüberprüfung und Typinferenz
- Korrektheitsargumentationen
- Funktionale Datenstrukturen
- Bedarfsauswertung und unendliche Strukturen
- Typklassen
- Funktoren und Monaden
- Beispielanwendungen

Zusätzlich werden in der Übung praxisrelevante Aspekte der Anwendungsentwicklung mit der Programmiersprache Haskell behandelt, die nicht Bestandteil der Vorlesung sind. Die Bearbeitung der Übungsaufgaben folgt parallel zum Stoff der Vorlesung in Zweiergruppen.

### Qualifikationsziele:

Die Studierenden ...

- beherrschen fortgeschrittene Techniken der funktionalen Programmierung am Beispiel der Sprache Haskell.
- können mit Funktionen höherer Ordnung, mit polymorphen Datentypen und Typklassen, mit Funktoren, Monaden, Monoiden und weiteren mathematischen Strukturen umgehen und in neuen Kontexten anwenden.
- beherrschen Fähigkeiten in der Modellbildung und Abstraktion und können sie anwenden.
- kennen die Bezüge zwischen Mathematik und funktionaler Programmierung.
- kennen die Vor- und Nachteile des funktionalen Paradigmas für Anwendungen der IT-Sicherheit.

### Verwendbarkeit:

Das Modul kann sinnvoll im Modul "Methoden der Künstlichen Intelligenz", in Projekten und der Master-Thesis genutzt werden.

### Voraussetzungen und Empfehlungen:

Voraussetzungen sind Kenntnisse und praktische Erfahrungen in höheren Programmiersprachen, insbesondere mit getypten Sprachen. Außerdem werden Kenntnisse über Diskrete Mathematik und algebraische Strukturen erwartet.

Elementares Wissen über Komplexitätstheorie wird ebenfalls vorausgesetzt.

Zudem ist die Installation von Haskell auf dem eigenen Rechner empfehlenswert.

### Literatur:

- Miran Lipovaca: Learn You a Haskell for Great Good!: A Beginner's Guide, No Starch Press, 2011, ISBN: 978-1593272838
- Graham Hutton: Programming in Haskell, Cambridge University Press, 2016, ISBN: 978-1316626221
- Richard Bird: Introduction to Functional Programming using Haskell, 2nd Edition Prentice Hall, New Jersey, 1998, ISBN: 978-0134843469

### Studiengänge:

- Informatik Master of Science (1. Semester)
- IT-Sicherheit Master of Science (1. Semester)

## ◆ MM006 - Learning and Softcomputing

Verantwortliche:	Christian-Arved Bohn
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

### Bestandteile:

Teilleistung	Lernform	Prüfungsform	-umfang	-dauer	ECTS	Benotung	Turnus	Aufwand	Lehrende
TM001 - Learning and Softcomputing	Workshop	Portfolio-Prüfung	25 Seiten	15 Min.	5.0	Drittelpnoten	jährlich	150 Stunden	Ulrich Hoffmann

### Lehrinhalte:

- Einführung, Motivation
- Maschinelles Lernen
- Das Konzept der Neuronalen Netze
  - Grundprinzip
  - Arten von Neuronalen Netzen
  - Einlagige Neuronale Netze
  - Mehrlagige Netze
  - Ein Lernverfahren: Backpropagation
- Das Konzept der Support Vector Machines
  - Grundlagen und Eigenschaften
  - Klassifikation durch Hyperebenen
  - Der Kernel-Trick
  - Aspekte der Implementierung von SVM
- Praktische Projektarbeit in Gruppen zur eigenständigen Implementierung und Untersuchung eines ausgewählten Themenkomplexes.
- Regelmäßige Diskussion der Ergebnisse der Projektarbeit und gruppenweise Abschlusspräsentation.

### Qualifikationsziele:

Die Studierenden ...

- besitzen grundlegende Kompetenz zum Verständnis für lernfähige, fehlertolerante Problemlösungsansätze.
- haben die Fähigkeit zur Erkennung und Unterscheidung verschiedener maschineller Lernverfahren und Verarbeitungskonzepte.
- haben grundlegendes Verständnis der Themenkomplex Künstlicher Neuronaler Netze (KNN) sowie der Support Vector Machines (SVM)
- besitzen die Fähigkeit unterschiedlichen Ansätze überwachter und unüberwachter Klassifikationsverfahren und ihre mathematischen Hintergründe zu durchdringen.
- haben die Fähigkeit, eine beispielhafte Implementierung dargestellten theoretischen Konzepten im Rahmen selbständiger, gruppenorientierter Projektarbeit gezielt und strukturiert umzusetzen.
- besitzen die Fähigkeit die von ihnen im Rahmen der Projektarbeit erarbeiteten Sachverhalte zu kondensieren und in angemessenen Vortragsstil und geeigneter Präsentationstechniken nachvollziehbar dazustellen. In freier Diskussion können sie sich über komplexe wissenschaftlichen Sachverhalte auseinandersetzen.
- besitzen vertiefte Kenntnisse des Themas der konkret bearbeiteten Machine-Learning-Aufgabe, also etwa zu Bildverarbeitung, IT-Sicherheit, E-Commerce oder Betriebswirtschaftslehre.

### Verwendbarkeit:

Das Modul ist sinnvoll mit dem Modul "Robotics" und den grundlegenden Modulen "Einführung in die Robotik" und "Bildbearbeitung und -analyse" kombinierbar. Zudem bietet sich ein Zusammenspiel in Richtung Data Sciences an, wenn es mit den grundlegenden Modulen "Deskriptive Statistik & Grundlagen der Linearen Algebra", "Induktive Statistik" und im Master mit den Modulen "Business Intelligence", "Empirische Forschungs- und Analysemethoden" und "Entscheidungsunterstützung" kombiniert wird.

### Voraussetzungen und Empfehlungen:

- Vertrautheit mit den mathematischen Grundlagen und statistischen Konzepten, die für das Verständnis von maschinellem Lernen und neuronalen Netzen erforderlich sind.
- Fähigkeit zur Anwendung von mathematischen Methoden in der Analyse und Implementierung von Algorithmen.
- Kenntnisse in grundlegender Programmierung und Informatik, um die Implementierung von maschinellen Lernverfahren und neuronalen Netzen zu ermöglichen.

- Erfahrung im Umgang mit Programmiersprachen und Entwicklungsumgebungen, die für maschinelles Lernen und neuronale Netze relevant sind.
- Fähigkeit zur eigenständigen Bearbeitung und Implementierung komplexer Themenkomplexe im Rahmen von Projektarbeiten.
- Kompetenzen in der Zusammenarbeit und Kommunikation innerhalb von Gruppen sowie in der regelmäßigen Präsentation und Diskussion von Projektergebnissen.

### **Literatur:**

- Kecman: Learning and Softcomputing, MIT Press, 2001
- Nauck, Klawonn: Neuronale Netze und Fuzzy-Systeme, R. Kruse, Vieweg 1996
- Bishop: Neural Networks for Pattern Recognition, Oxford Press 1995
- Sutton, Barto: Reinforcement Learning: An Introduction, MIT Press, Cambridge, MA, 1998
- Christianini, Shawe-Taylor: Support Vector Machines, N., Cambridge Press, 2000
- Brause: Neuronale Netze, Teubner, 1991

### **Studiengänge:**

- Data Science & Artificial Intelligence Master of Science (1. Semester)
- Informatik Master of Science (1. Semester)
- IT-Sicherheit Master of Science (1. Semester)
- Wirtschaftsinformatik / IT-Management Master of Science (Wahlmöglichkeit 1. Semester)

# ◆ MM009 - Workshop Kryptographie

Verantwortliche:	Gerd Beuster
Moduldauer:	6 Monate
Unterrichtssprache:	english

## Bestandteile:

Teilleistung	Lernform	Prüfungsform	-umfang	-dauer	ECTS	Benotung	Turnus	Aufwand	Lehrende
TM030 - Workshop Kryptographie	Workshop	Abnahme	12 Aufgaben		5.0	Drittelnoten	jährlich	150 Stunden	Gerd Beuster

## Lehrinhalte:

- Theorie der Kryptographie
  - Semantische Sicherheit
  - Unbrechbare Verschlüsselung und One Time Pad
  - Diffusion und Konfusion
- Klassische Kryptographie
  - Substitution und Transposition
  - Affine Verschlüsselung
  - Rotormaschinen
- Moderne Kryptographie
  - Stream- und Block-Chiffren
  - DES und GOST
  - AES
- Blockchiffrierung Betriebsarten
  - ECB, CBC, CTR, AES-GCM
- Zufallszahlengeneratoren
  - TRNG und PRNG
  - Voraussetzungen für CSPRNG
- Hashing
  - Hashing-Algorithmen
    - SHA 2
    - SHA 3
  - Nachrichten-Authentifizierung
    - CMAC und HMAC
- Asymmetrische Kryptographie
  - Diffie-Hellman
  - Elliptische Kurven
  - Asymmetrische Verschlüsselung und digitale Signaturen
- Post-Quantum Cryptography
  - CRYSTALS-Kyber
- Hardware-Kryptographie
  - Trusted Computing
  - Smartcards
  - Differential Power Analysis

## Qualifikationsziele:

Nach Abschluss des Moduls sind die Studierenden in der Lage, ...

- Sicherheitswerkzeuge als wesentlichen Baustein moderner Informations- und Kommunikationssysteme zu verwenden.
- ihr Wissen über alle relevanten Aspekte der Daten-, Netzwerk- und Websicherheit anzuwenden.
- die Anwendung kryptographischer Methoden, insbesondere zur Authentifizierung, Verschlüsselung und Integritäts-erhaltung zu betrachten.
- die algorithmischen Stärken und Schwächen der kryptographischen Verfahren zu bewerten.
- kryptographische Protokolle zu bewerten und zu implementieren, insbesondere für die Authentifizierung im E-Commerce.
- alle für die Implementierung und Anwendung kryptographischer Methoden relevanten Nebenbedingungen zu berücksichtigen.
- die Qualität von Zufallszahlengeneratoren zu beurteilen.
- die Eignung von Software- und Hardware-Kryptographie für eine bestimmte Aufgabe abzuschätzen.

## Verwendbarkeit:

Die Studierenden erwerben fortgeschrittene Kenntnisse über die mathematischen Grundlagen der Kryptographie und deren praktische Anwendung. Diese Kenntnisse können in allen Bereichen eingesetzt



werden, in denen kryptographische Methoden verwendet werden.

### **Voraussetzungen und Empfehlungen:**

Für dieses Modul sind Grundkenntnisse der diskreten Mathematik erforderlich. Weiterhin müssen die Studierenden über grundlegende Kenntnisse der Programmierung verfügen, insbesondere in der Programmiersprache Python.

### **Literatur:**

- Bos, Joppe, et al. : CRYSTALS – Kyber: A CCA-secure module-lattice-based KEM. In: O'Conner, L. (ed.) : Proceedings of the 2018 IEEE European Symposium on Security and Privacy (EuroS&P). London, UK, 2018, pp. 353-367, doi: 10.1109/EuroSP.2018.00032.
- Ferguson, Niels; Schneier, Bruce; Kohno, Tadayoshi: Cryptography Engineering : Design Principles and Practical Applications. Indianapolis (IN), USA: Wiley Publishing, 2010.
- Katz, Jonathan; Lindell, Yehuda : Introduction to Modern Cryptography. Boca Raton (FL), USA: CRC Press, 2007.
- Mao, Wenbo: Modern Cryptography: Theory and Practice, Upper Saddle River (NJ), USA: Prentice Hall, 2003.
- NIST : SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. FIPS PUB 202, <https://doi.org/10.6028/NIST.FIPS.202>, 2015.
- Ristic, Ivan : Bulletproof TLS and PKI. 2. Edition. London, UK: Feisty Duck, 2022.
- Stallings, William: Cryptography and Network Security : Principles and Practice. 8. Edition. London, UK: Pearson, 2022.
- Stinson, Douglas R. : Cryptography : Theory and Practice. 4. Edition. Boca Raton (FL), USA: CRC Press, 2018.
- Swenson, Christopher: Modern Cryptanalysis : Techniques for Advanced Code Breaking. Indianapolis (IN), USA: Wiley Publishing, 2008.

### **Studiengänge:**

- Informatik Master of Science (1. Semester)
- IT-Sicherheit Master of Science (1. Semester)
- IT Engineering Master of Science (Wahlmöglichkeit 1. Semester)

## ◆ MM018 - Robotics

Verantwortliche:	Ulrich Hoffmann
Moduldauer:	6 Monate
Unterrichtssprache:	english

### Bestandteile:

Teilleistung	Lernform	Prüfungsform	-umfang	-dauer	ECTS	Benotung	Turnus	Aufwand	Lehrende
TM032 - Robotics	Vorlesung mit integrierter Übung	Portfolio-Prüfung	25 Seiten	20 Min.	5.0	Drittelnoten	jährlich	150 Stunden	Ulrich Hoffmann

### Lehrinhalte:

- Aufbau und Zusammensetzung von Robotern
  - Kinematik
  - Bewegung und Beweger
  - Effektoren
  - Programmier-Systeme
- Bewegungsmodellierung
  - Punkt-zu-Punkt-Steuerung
  - Interpolation von Trajektorien
- Modellierung von Aktionen
- Intelligente Sensoren
  - Taktile Sensoren
  - Optical sensors
- Lernende Roboter
- Praktisches Projekt
  - Eigenverantwortliches Umsetzen eines Projektes innerhalb des komplexen Themengebiets
  - Experimentelles Erforschen neuer Ansätze und Ideen, welche über den Vorlesungsinhalt hinaus gehen
  - Regelmäßige Diskussion der Projektergebnisse und Präsentationen vor allen Gruppen

### Qualifikationsziele:

#### Studierende

- verfügen über Grundkenntnisse ausgewählter Konzepte und Technologien der Robotik.
- verstehen vor allem die Eigenschaften mobiler autonomer Systeme gründlich.
- haben ein tiefes Verständnis der technischen Grundlagen der Robotik und insbesondere der Konzepte der Bewegungs- und Aktionsmodellierung sowie intelligenter lernender Sensoren als Grundlage des autonomen Roboterverhaltens.
- sind in der Lage, exemplarische Implementierungen der vorgestellten theoretischen Konzepte in einem selbstorganisierten und gruppenorientierten Projekt zu realisieren.
- können ausgehend von den vorgestellten Konzepten selbstständig neue Lösungsansätze entwickeln, umsetzen und das Ergebnis beurteilen.
- haben die Kompetenz, praktische Probleme zu verstehen, die auftreten, wenn Roboteraktionen durch visuelle Bilder gesteuert werden.
- sind in der Lage, ihre wissenschaftlichen Ergebnisse in einer geeigneten Präsentation mit geeigneten Präsentationstechniken verständlich zu vermitteln.
- sind in der Lage, komplexe wissenschaftliche Sachverhalte in einem Fachgespräch kompetent zu vermitteln.

### Verwendbarkeit:

Das Modul ist sinnvoll kombiniert mit den Basismodulen "Einführung in die Robotik" und "Bildverarbeitung und -analyse" sowie dem Modul "Lernen & Softcomputing". Es kann in allen technischen Studiengängen eingesetzt werden.

### Voraussetzungen und Empfehlungen:

- Kenntnisse in Programmier-Systemen und der Entwicklung imperativer Programme
- Fähigkeit zur eigenverantwortlichen Umsetzung eines Projektes innerhalb des komplexen Themengebiets der Robotik.
- Kompetenzen im experimentellen Erforschen neuer Ansätze und Ideen, die über den Vorlesungsinhalt hinausgehen, sowie in der regelmäßigen Diskussion und Präsentation von Projektergebnissen.
- Kenntnisse in Linearer Algebra

- Grundlegende Fähigkeiten imperative Programme zu erstellen und auf Software-Bibliotheken zuzugreifen
- Grundkenntnisse und grundlegende Fähigkeiten in der Programmierung von Bildverarbeitungsalgorithmen und der Benutzung einschlägiger Bibliotheken.

### **Literatur:**

- Blume, Dillmann: Frei Programmierbare Roboter, Vogel Verlag, 1981
- McKerrow: Introduction to Robotics: Introduction to Robotics, Addison Wesley, 1991
- Stienecker: The KUKA Robot Programming Language, Eigenverlag, 2011
- Vukobratovic: Introduction to Robotics, Springer, 1989

### **Studiengänge:**

- Data Science & Artificial Intelligence Master of Science (Wahlmöglichkeit 1. Semester)
- Informatik Master of Science (Wahlmöglichkeit 1. Semester)
- IT Engineering Master of Science (Wahlmöglichkeit 1. Semester)

## ◆ MM023 - Seminar Informatik

Verantwortliche:	Ulrich Hoffmann
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

### Bestandteile:

Teilleistung	Lernform	Prüfungsform	-umfang	-dauer	ECTS	Benotung	Turnus	Aufwand	Lehrende
TM024 - Seminar	Seminar	Schriftl. Ausarbeitung (ggf. mit Präsentation)	15 Seiten	60 Min.	5.0	Drittelnoten	jährlich	150 Stunden	Ulrich Hoffmann

### Lehrinhalte:

Fachvorträge mit anschließender Gruppendiskussion.

### Qualifikationsziele:

Die Studierenden ...

- sind in der Lage, eine wissenschaftliche fundierte Lösung für theoretische und/oder praktische Problemstellungen primär aus dem Themengebiet sowie ähnlichen Gebieten zu entwickeln.
- zeigen eine verbesserte Problemlösungstechnik, sicherere Verwendung von Termini, präzise Strukturierung im Aufbau schriftlicher Arbeiten und Einhalten der Formalia.
- zeigen eine auf Masterniveau angemessene Vortragstechnik im Rahmen der Präsentation der Ergebnisse.

### Verwendbarkeit:

Die Fähigkeit, theoriegestützt zu arbeiten, wird in der Master-Thesis benötigt.

### Voraussetzungen und Empfehlungen:

- Fähigkeit, wissenschaftliche Themen zu strukturieren und eine eigenständige Zielsetzung zu entwickeln.
- Kenntnisse im Recherchieren und Aufbereiten wissenschaftlicher Inhalte.
- Vertrautheit mit den formalen Anforderungen und Kriterien wissenschaftlicher Arbeiten.
- Fähigkeit, formale Richtlinien sicher zu beachten und anzuwenden.
- Fähigkeit, schriftliche Ausarbeitungen größeren Umfangs zu erstellen.
- Kompetenzen im Vortragen und Diskutieren der Ergebnisse mit anderen Seminarteilnehmern.

### Literatur:

Recherche nach aufgabenbezogener Literatur, teilweise aufgabenspezifische Vorgabe einzelner Literaturquellen.

### Studiengänge:

- Informatik Master of Science (1. Semester)

## ◆ MM062 - Praktikum Virtuelle Realität und Simulation

Verantwortliche:	Christian-Arved Bohn
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

### Bestandteile:

Teilleistung	Lernform	Prüfungsform	-umfang	-dauer	ECTS	Benotung	Turnus	Aufwand	Lehrende
TM038 - Prakt. Virtuelle Realität und Simulation	Projektarbeit	Abnahme	15 Aufgaben	20 Min.	5.0	Drittelnoten	jährlich	150 Stunden	Christian-Arved Bohn

### Lehrinhalte:

Programmierung in C. Einarbeitung in das Labor-Framework zur Verwendung des CAVE. Entwicklung einer Projektaufgabenstellung und Durchführung des Projektes zu Themengebieten der fortgeschrittenen Virtuellen Realität. Praktikumsbegleitend finden kleine Vorlesungseinheiten zu bestimmten Themen, die für konkrete Projekte gebraucht werden, statt.

### Qualifikationsziele:

Durch eine in kleinen Gruppen entwickelte und durchgeführte Projektaufgabe erlangen Studierende Kenntnisse über Spezialalgorithmen der Virtuellen Realität, die für gewöhnlich in vorgefertigten Softwaretools verborgen sind (z. B. Kalibrierung, Verarbeitung von Videobildern oder 3D-Klang). Auf diese Weise kann ein tiefgehendes Verständnis über typische VR-Installationen erlangt werden.

### Verwendbarkeit:

Das Modul bereichert Erfahrungen aus Modulen der Softwareentwicklung und diverser Praktika und dient vor allem der weiteren Vorbereitung auf das Betriebspraktikum in einem externen Unternehmen.

### Voraussetzungen und Empfehlungen:

Grundlagen der Mathematik, Voresung "Programmstrukturen 1", Vorlesung "Virtual and Augmented Reality"

### Literatur:

- Doug A. Bowman, Ernst Kruijff, Joseph J. Laviola: 3D User Interfaces: Theory and Practice, Addison-Wesley Longman, 2004.
- Ralf Dörner, et al.: Virtual und Augmented Reality (VR/AR): Grundlagen und Methoden der Virtuellen und Augmentierten Realität, Springer Vieweg, 2013.

### Studiengänge:

- Informatik Master of Science (Wahlmöglichkeit 1. Semester)

# ◆ MM162 - Moderne Software-Architekturen

Verantwortliche:	Ulrich Hoffmann
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

## Bestandteile:

Teilleistung	Lernform	Prüfungsform	-umfang	-dauer	ECTS	Benotung	Turnus	Aufwand	Lehrende
TM039 - Moderne Software-Architekturen	Vorlesung	Portfolio-Prüfung	25 Seiten	30 Min.	5.0	Drittelnoten	jährlich	150 Stunden	Ulrich Hoffmann

## Lehrinhalte:

- Virtualisierung
- Container
- Cloud Computing
- Serverless Computing
- grundlegende verteilte Algorithmen
  - Konsensalgorithmen,
  - Synchronisationsalgorithmen
- Architekturmuster
  - Map/Reduce
  - Service Orientierte Architektur
  - Microservices
  - Pipelines
    - AI Pipelines
    - Deployment Pipelines
    - DevOps Pipelines
- Seminaristischer Unterricht
  - Kurzvortrag zu ausgewähltem Grundlagenthema
  - praktisches Projekt zu ausgewählter Aufgabenstellung
    - regelmäßige Projektbesprechungen
    - Wissenschaftlicher Vortrag zur Vorstellung der Projektergebnisse

## Qualifikationsziele:

Die Studierenden ...

- besitzen eingehende Kenntnisse über Struktur, Entwicklung, Bereitstellung und Betrieb großer Software-Systeme.
- haben die Fähigkeit umfangreiche Softwaresysteme bereitzustellen.
- haben die Fähigkeit eigene Softwaresysteme bereitzustellen.
- besitzen Kenntnisse und Fähigkeiten, große Softwaresystem automatisch überwachen zu lassen und Messungen an ihnen durchzuführen.
- Sie besitzen grundlegende Fähigkeiten aufgrund von Messergebnissen eine Optimierung und Skalierung von Softwaresystemen durchzuführen.
- haben die Fähigkeit, für ein Softwaresystem Anforderungen zu erheben, es geeignet zu strukturieren und auf angemessene Weise in Betrieb zu nehmen.
- besitzen die Fähigkeit ihr Vorgehen kritisch zu hinterfragen und nachvollziehbar zu kommunizieren.
- besitzen die Fähigkeit die von ihnen im Rahmen der Projektarbeit erarbeiteten Sachverhalte zu kondensieren und in angemessenen Vortragstil und geeigneter Präsentationstechniken nachvollziehbar dazustellen. In freier Diskussion können sie sich über komplexe wissenschaftliche Sachverhalte der großen Softwaresysteme und ihres Betriebs auseinandersetzen.

## Verwendbarkeit:

Das Modul ist sinnvoll mit den Modulen "Distributed Systems" und "Learning and Softcomputing" kombinierbar.

## Voraussetzungen und Empfehlungen:

- Erfahrung in der imperativen und objekt-orientierten Programmierung
- grundlegendes Verständnis der Sicherheitsanforderungen in der Informationstechnik
- Fähigkeit zur Implementierung von Sicherheitsmaßnahmen
- Kenntnisse über die Funktionsweisen von Rechnernetzen und des World Wide Webs / Internets.

## Literatur:

- Bass, Weber, Zhu: DevOps: A Software Architect's Perspective, Addison Wesley 2015
- Bass, Kazman, Clements: Software Architecture in Practice, Addison Wesley, 2012
- Fowler: Patterns of Enterprise Application Architecture, Addison Wesley, 2002
- Hapke, Nelson: Building Machine Learning Pipelines, O'Reilly Media, 2020
- Humble, Farley: Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation, Addison-Wesley 2010
- Josuttis: SOA in Practice: The Art of Distributed System Design, O'Reilly Media 2007
- Kim, Willis, Debois, Humble: The DevOPS Handbook, IT Revolution Press, 2016
- Krafzig, Banke, Slama: Enterprise SOA: Service-Oriented Architecture Best Practices, Prentice Hall 2004
- Newman: Monolith to Microservices, O'Reilly Media, 2019
- Saito, Lee, Wu: DevOps with Kubernetes, Packt Publishing, 2019

## Studiengänge:

- Informatik Master of Science (Wahlmöglichkeit 1. Semester)

# ◆ MM010 - Aktuelle Entwicklungen in der Informatik

Verantwortliche:	Ulrich Hoffmann
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

## Bestandteile:

Teilleistung	Lernform	Prüfungsform	-umfang	-dauer	ECTS	Benotung	Turnus	Aufwand	Lehrende
TM031 - Workshop Aktuelle Entwicklungen in der Informatik	Workshop	Assessment	25 Seiten	30 Min.	5.0	Drittelnoten	jährlich	150 Stunden	Frank Huch

## Lehrinhalte:

themenabhängig

## Qualifikationsziele:

Studierende besitzen ...

- die Fähigkeit sich intensiv mit jeweils aktuellen Themen der theoretischen, praktischen, angewandten Informatik auseinanderzusetzen.
- Kenntnisse ausgewählter, fortgeschrittenen, modernen Informatik-Themen. Sie kennen im Detail die im jeweiligen Themengebiet relevanten Fragestellungen und können Lösungsansätze im Hinblick auf ihre Eignung bewerten und beurteilen.
- die Fähigkeiten einschlägige Softwaresysteme der jeweiligen Themenstellung zu bewerten und einzusetzen.

## Verwendbarkeit:

Das Modul ist mit den Modulen "Distributed Systems" und "Funktionale Programmierung" aber auch mit dem Modul "Seminar Master" gut kombinierbar.

## Voraussetzungen und Empfehlungen:

Grundlegende Fertigkeiten der Online-Recherche einschließlich des Beurteilens von relevanten Trends mit Abgrenzung zum Buzzword-Bingo ist hilfreich. Das Lesen und Verstehen von Fachliteratur in englischer Sprache ist notwendig.

## Literatur:

themenabhängig

## Studiengänge:

- Informatik Master of Science (Wahlmöglichkeit 2. Semester)



# ◆ MM027 - Konzepte der Datenbanktechnologie

Verantwortliche:	Ulrich Hoffmann
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

## Bestandteile:

Teilleistung	Lernform	Prüfungsform	-umfang	-dauer	ECTS	Benotung	Turnus	Aufwand	Lehrende
TM002 - Konzepte der Datenbanktechnologie	Vorlesung	Klausur / Mündliche Prüfung		60 Min.	3.0	Drittelnoten	jährlich	90 Stunden	Michael Predeschly
TM003 - Ubg. Konzepte der Datenbanktechnologie	Übung	Abnahme		10 Min.	2.0	Bestanden/nicht Bestanden	jährlich	60 Stunden	Tim Wetzel

## Lehrinhalte:

- Grundlagen Datenbanksysteme
  - Persistenz
  - Transaktionen
  - 2PL
  - Datenschutz und Datensicherheit
- Objekt-relationales Mapping
  - Java Persistence API (JPA)
- NoSQL-Datenbanksysteme
  - Verteilte Wert/Schlüssel-Speicher
  - Dokumentendatenbanken
  - Graph-Datenbanken
- Verteilung von Daten

Vorlesungsbegleitende praktische Übungen zu Objektrelationalem Mapping und anderen alternativen Persistenzansätzen.

Erstellung einer NoSQL-Datenbank mit einem kompletten CRUD-Zyklus.

## Qualifikationsziele:

Studierende ...

- beherrschen die Fähigkeit Objektrelationales Mapping anzuwenden bzw. in Betrieb zu nehmen und es zur Lösung von Problemen einzusetzen.
- sind mit den praktisch auftretenden Schwierigkeiten vertraut und können sie systematisch überwinden.
- sind in der Lage eine NoSQL-Datenbank einzurichten, sie mit Daten zu füllen und anfragen an sie zu stellen

Die Studierenden erlangen die ...

- Kenntnis, der für die Implementierung von Datenbanksystemen wichtigen Architekturprinzipien, Datenstrukturen und Algorithmen und damit Kenntnis des Aufbaus und der internen Arbeit eines großen komplexen Softwaresystems.
- Fähigkeit, die Arbeitsweise von Datenbanksystemen zu optimieren bzw. selbst Architekturen für große komplexe Softwaresysteme zu entwerfen.
- Fähigkeiten eines Datenbankadministrators für Datenbanksysteme.
- Konzepte und Techniken des Datenschutzes, als auch der Datensicherheit

## Verwendbarkeit:

Das Modul ist sinnvoll im Datenbanken-Curriculum zusammen mit den grundlegenden Modulen "Einführung in Datenbanken" und "Datenbanktheorie und -implementierung" aber auch den Programmierintroduktionsmodulen ("Einführung in die Programmierung", "Programmstrukturen 1") zu kombinieren. Auch eine Kombination mit dem grundlegenden Modul "Systemmodellierung" ist ratsam.

## Voraussetzungen und Empfehlungen:

- Vertrautheit mit den grundlegenden Konzepten von Datenbanksystemen, einschließlich der Prinzipien der Persistenz und Transaktionen.
- Verständnis der relationalen Datenbankmodelle und Kenntnisse in SQL.
- Kenntnisse in der objekt-orientierten Programmierung

## Literatur:

- siehe Vorlesung
  - diverse Online-Quellen
- 
- KEMPER, Alfons; EICKLER, Andre:  
Datenbanksysteme - Eine Einführung. Oldenbourg Verlag, 2004
  - KEITH, Mike; SCHINCARIOL, Merrik:  
Pro JPA 2 - Mastering the Java Persistence API. APress, 2009
  - BAUER, Christian; KING, Gavin:  
Java Persistence with Hibernate,  
Manning, Greenwich, 2007
  - SQL- & NoSQL-Datenbanken – Andreas Meier, Michael Kaufmann; eXamen.press Springer Vieweg
  - Sieben Wochen, sieben Datenbanken – Eric Redmond, Jim R. Wilson; O'Reilly
  - NoSQL for Dummies, Adam Fowler; For Dummies-Verlag
  - div. Konferenzbeiträge und Forschungsarbeiten zu moderneren Entwicklungen der Datenbanktechnologie

## Studiengänge:

- Data Science & Artificial Intelligence Master of Science (Wahlmöglichkeit 2. Semester)
- Informatik Master of Science (2. Semester)
- IT-Sicherheit Master of Science (2. Semester)
- Wirtschaftsinformatik / IT-Management Master of Science (Wahlmöglichkeit 2. Semester)

## ◆ MM029 - Berechenbarkeit und Verifikation

Verantwortliche:	Sebastian Iwanowski
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch/englisch

### Bestandteile:

Teilleistung	Lernform	Prüfungsform	-umfang	-dauer	ECTS	Benotung	Turnus	Aufwand	Lehrende
TM033 - Berechenbarkeit und Komplexität, Formale Spezi...	Vorlesung	Klausur / Mündliche Prüfung		120 Min.	5.0	Drittelnoten	Wintersemester	150 Stunden	Sebastian Iwanowski Ulrich Hoffmann

### Lehrinhalte:

- Berechenbarkeit und Nichtberechenbarkeit
    - Präzisierung der Begriffe Problem und Algorithmen für die Theorie der Berechenbarkeit
    - Turingmaschinen im Detail
    - Komplexitätsklassen für Turingmaschinen
    - Beispiele für unentscheidbare Probleme
    - Beweise der Unentscheidbarkeit für ausgewählte Probleme
  - NP-vollständige Probleme
    - Historie des P-NP-Problems
    - Beweis der NP-Vollständigkeit von SATISFIABILITY
    - Übersicht über NP-vollständige Probleme
    - Reduktionsmethode zum Beweis von NP-Vollständigkeit mit Beispielen
  - Optimierungsaufgaben für NP-vollständige Probleme
    - Lösungstechniken für NP-vollständige Probleme
    - Übersicht über wichtige Anwendungen - Vergleich zu Verfahren der Künstlichen Intelligenz
- 
- Mathematische und logische Grundlagen der Spezifikation und Verifikation; Mengen, Multimengen, Verbände, partielle und totale Funktionen, algebraische Strukturen, Aussagen- und Prädikatenlogik, Modallogik, temporale Logik
  - Algebraische Spezifikation; Terme, Gleichungen; Fallbeispiel einer algebraischen Spezifikation; Datenstrukturen, Operationen, Nachweis von Eigenschaften; maschinenunterstütztes Beweisen von Eigenschaften
  - Modellorientierte Spezifikation; Fallbeispiel einer modellorientierten Spezifikation
  - Konstruktion korrekter Programme aus Spezifikationen
  - Aktuelle Spezifikationssprachen im Überblick

### Qualifikationsziele:

Nach Abschluss der Veranstaltung besitzen die Studierenden folgende Kompetenzen:

- Fundierter theoretischer Überblick über die Möglichkeiten des Problemlösens.
- Theoretisch fundierte Kenntnis der Grenzen der Berechenbarkeit und der effizienten Lösbarkeit.
- Kenntnis der Alternativen für die Praxis bei theoretisch unbefriedigenden Resultaten.

Die Studierenden ...

- erlangen fundierte Kenntnisse der mathematischen Grundlagen der formalen Spezifikation und Verifikation.
- beherrschen verschiedene Spezifikationsstile.
- bekommen einen Einblick in verschiedene formale Spezifikationssprachen.
- erlangen die Fähigkeit, Spezifikationen systematisch zu konstruieren.
- können mathematische Beweise von Eigenschaften spezifizierte Software-Systeme führen.
- erlangen grundlegende Kenntnisse der Verifikation mit automatischen Beweissystemen.
- sind mit der Spezifikation von Sicherheitsbedingungen vertraut.
- können beurteilen, wie sich formale Methoden auf die IT-Sicherheit auswirken.

### Verwendbarkeit:

Das Modul gibt eine Vertiefung der wissenschaftlichen Grundlagen des Informatikstudiums. Es ergänzt auf diese Weise das grundlegendere und anwendungsbezogenere Modul "Algorithmics", setzt dieses aber nicht voraus. Für IT-Sicherheitsapplikationen liefert es die theoretische Grundlage.

## Voraussetzungen und Empfehlungen:

Diskrete Mathematik aus dem Bachelorstudium, im Besonderen die Mengenlehre und Beweise

## Literatur:

- [Garey, Michael R.; Johnson, David S. \(1979\), Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, ISBN 0-7167-1045-5](#)
  - Hopcroft, John E.; Motwani, Rajeev; Ullman, Jeffrey D.:  
Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie.  
2. überarb. Aufl. München: Addison-Wesley Longman Verlag, 2002.
  - Vossen, Gottfried; Witt, Kurt-Ulrich:  
Theoretische Informatik.  
Braunschweig: Verlag Vieweg & Teubner 2004 (3. Auflage), ISBN 978-3528231477
  - Wagenknecht, C.:  
Algorithmen und Komplexität,  
Fachbuchverlag Leipzig 2003
  - Winter, R.:  
Theoretische Informatik,  
Oldenbourg-Verlag München 2002
- 
- BJØRNER, Dines:  
Software Engineering 1.  
Heidelberg: Springer Verlag, 2006
  - DILLER, Antoni:  
Z An Introduction to Formal Methods.  
New York: Wiley & Sons, 1994
  - EHRICH/GOGOLLA/LIPECK:  
Algebraische Spezifikation abstrakter Datentypen.  
Stuttgart: Teubner Verlag, 1989
  - GOOS, Gerhard:  
Vorlesungen über Informatik Band 1 - Grundlagen und funktionales Programmieren.  
Heidelberg: Springer Verlag, 2005
  - LAMPORT, Leslie:  
Specifying Systems.  
Amsterdam: Addison-Wesley, 2002
  - SCHÖNING, Uwe:  
Logik für Informatiker.  
Heidelberg: Spektrum Akademischer Verlag, 2000
  - WORDSWORTH, J., B.:  
Software Development with Z.  
New York: Addison-Wesley, 1992

## Studiengänge:

- Informatik Master of Science (2. Semester)
- IT-Sicherheit Master of Science (2. Semester)

# ◆ MM033 - Methoden der Künstlichen Intelligenz

Verantwortliche:	Gerd Beuster
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch/englisch

## Bestandteile:

Teilleistung	Lernform	Prüfungsform	-umfang	-dauer	ECTS	Benotung	Turnus	Aufwand	Lehrende
TM005 - Methoden der Künstlichen Intelligenz	Vorlesung	Klausur / Mündliche Prüfung		90 Min.	5.0	Drittelnoten	jährlich	150 Stunden	Gerd Beuster

## Lehrinhalte:

- Einführung in die Künstliche Intelligenz
- Intelligente Agenten
- Suchverfahren
- Aussagenlogik
- Logikbasierte autonome Agenten
- Prädikatenlogik
- Formale Logik und Sprachmodelle
- Grenzen der Prädikatenlogik
- Logikprogrammierung
- Knowledge Graphs

## Qualifikationsziele:

Die Studierenden sind in der Lage, Probleme der realen Welt in die Formalismen der klassischen Logiken (Aussagen- und Prädikatenlogik) umzusetzen. Sie kennen die Syntax und Semantiken der klassischen Logiken und die Grenzen der formallogischen Beweisbarkeit. Sie sind mit Methoden des automatischen Schließens vertraut. Die Studierenden können große Sprachmodelle (LLMs) in Kombination mit formallogischen Methoden nutzen.

## Verwendbarkeit:

Die im Modul erworbenen Fähigkeiten können überall dort verwendet werden, wo autonom handelnde Agenten benötigt werden.

## Voraussetzungen und Empfehlungen:

Das Modul setzt voraus, dass die Studierenden die grundlegenden Algorithmen der Informatik und Grundlagen diskreter algebraischer Strukturen kennen.

## Literatur:

- Harrison, John: Handbook of Practical Logic and Automated Reasoning, Cambridge: Cambridge University Press, 2009.
- Mackworth, Alan K.; Poole, David: Artificial Intelligence : Foundations of Computational Agents. 3. Auflage. Cambridge: Cambridge University Press, 2023.
- Norvig, Peter; Russell, Stuart: Artificial Intelligence : A Modern Approach. 4. Auflage. München: Pearson Deutschland GmbH, 2021.
- Schöning, Uwe: Logik für Informatiker, 5. Auflage. Heidelberg: Spektrum Akademischer Verlag, 2000.
- Lipovaca, Miran: Learn You a Haskell for Great Good! San Francisco (CA), USA: No Starch Press, 2012.
- Blackburn, Patrick; Bos, Johan; Striegnitz, Kristina: Learn Prolog Now!. London, UK: College Publications, 2006.

## Studiengänge:

- Data Science & Artificial Intelligence Master of Science (2. Semester)
- Informatik Master of Science (2. Semester)
- Wirtschaftsinformatik / IT-Management Master of Science (2. Semester)

## ◆ MM035 - Distributed Systems

Verantwortliche:	Ulrich Hoffmann
Moduldauer:	6 Monate
Unterrichtssprache:	english

### Bestandteile:

Teilleistung	Lernform	Prüfungsform	-umfang	-dauer	ECTS	Benotung	Turnus	Aufwand	Lehrende
TM006 - Distributed Systems	Vorlesung	Klausur / Mündliche Prüfung		60 Min.	3.0	Drittelnoten	jährlich	90 Stunden	Ulrich Hoffmann
TM007 - Tutorial: Distributed Systems	Übung	Abnahme	10 Aufgaben	5 Min.	2.0	Bestanden/nicht Bestanden	jährlich	60 Stunden	Ulrich Hoffmann

### Lehrinhalte:

- Praktische Beispiele
- Allgemeine Anforderungen an verteilte Systeme
- Die Client-Server-Beziehung und daraus resultierende Fragen
- Kommunikation in verteilten Systemen
- Dienste benennen
- Techniken für Gleichzeitigkeit
- Ferngespräche
- Alternative Paradigmen (Akteurskonzept, ...)
- Synchronisierung von Daten und Prozessen
- Koordinationsmethoden
- Replikationstechniken
- WEB-Dienste mit SOAP und REST
- Fehlertoleranzkonzepte
- Sicherheit in verteilten Systemen
- Programmierung mit Threads
- Kommunikation über Sockets, Struktur von Clients und Servern
- Ferner Prozeduraufruf / entfernter Methodenaufruf
- Verwendung von Benennungsdiensten
- Programmierung von WEB-Diensten (SOAP, Server/Client, WSDL, Datenbindung)
- verteiltes Programmieren mit alternativen Konzepten
- Programmierung von Synchronisierungsalgorithmen
- Programmierung verteilter Wahlalgorithmen
- Programmierung von REST-basierten Dienstleistungen und Kunden
- Fehlertolerante Programmierung in verteilten Systemen

Vorlesung mit begleitenden praktischen Übungen zur Programmierung verteilter Systeme und ihrer Algorithmen in verschiedenen Programmierparadigmen.

### Qualifikationsziele:

Die Studierenden gewinnen ...

- gründliches Verständnis der Prinzipien verteilter Anwendungen.
- Kenntnisse in der Beherrschung von Basistechnologien und aktuellen Software-Werkzeugen für verteilte Systeme.
- Zustandskenntnis der sind in verschiedenen Anwendungsbereichen wie Dienstleistungsvermittlung und E-Commerce.
- Kenntnisse über IT-Sicherheitsfragen in verteilten Systemen sowie über verschlüsselter Kommunikation.
- Kenntnisse der grundlegenden Algorithmen in verteilten Systemen.
- genaue Kenntnis der aktuellen Web-Service-Architekturen.
- praktische Fähigkeiten zur Realisierung eines Projekts.
- verteilte Programmierkenntnisse in verschiedenen Paradigmen.

Die Studenten ...

- erlangen die Fähigkeit, typische Softwaresysteme (Middleware) im Bereich der verteilten Systeme zu bedienen und zur Problemlösung einzusetzen.
- sind an Probleme gewöhnt, die in der Realität auftreten, und in der Lage, diese zu überwinden.
- haben einige praktische Erfahrungen mit IT-Sicherheitsfragen.
- wissen, wie man Verschlüsselung in verteilten Umgebungen einsetzt.
- eignen sich durch praktische Erfahrung ein tiefes Wissen über die spezifischen Eigenschaften verteilter Systeme an. Sie können diese Eigenschaften kategorisieren und bewerten.

## Verwendbarkeit:

Das Modul kann gut mit den Modulen "Funktionales Programmieren" und "Aktuelle Entwicklungen in der Informatik" sowie mit dem "Seminar-Master" kombiniert werden..

## Voraussetzungen und Empfehlungen:

- Erfahrung in der imperativen und objekt-orientierten Programmierung
- grundlegendes Verständnis der Sicherheitsanforderungen in der Informationstechnik
- Fähigkeit zur Implementierung von Sicherheitsmaßnahmen
- Kenntnisse über die Funktionsweisen von Rechnernetzen und des World Wide Webs / Internets.

## Literatur:

- siehe Vorlesung
  - Zahlreiche Online-Ressourcen
- 
- ARMSTRONG, Joe:  
Programming Erlang.  
Pragmatic Programmers, 2007
  - ODESKY, Martin; SPOON, Lex; VENNERS, Bill:  
Programming in Scala.  
Artima Press, Mountain View, 2008
  - COULOURIS, George; DOLLIMORE, Jean; KINDBERG, Tim:  
Distributed Systems, Concepts and Design.  
Addison-Wesley, 2011, ISBN 0-1321-4301-1
  - TANENBAUM, Andrew; VAN STEEN, Marten:  
Distributed Systems, Principles and Paradigms.  
Prentice Hall, 2006, ISBN 0-1323-9227-5

## Studiengänge:

- Informatik Master of Science (2. Semester)
- IT-Sicherheit Master of Science (2. Semester)
- IT Engineering Master of Science (2. Semester)
- Wirtschaftsinformatik / IT-Management Master of Science (Wahlmöglichkeit 2. Semester)

# ◆ MM042 - Digitale Kommunikationssysteme und Reconfigurable Computing

Verantwortliche:	Sergei Sawitzki
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

## Bestandteile:

Teilleistung	Lernform	Prüfungsform	-umfang	-dauer	ECTS	Benotung	Turnus	Aufwand	Lehrende
TM034 - Digitale Kommunikationssysteme	Vorlesung	Mündliche Prüfung		30 Min.	1.0	Drittelnoten	jährlich	30 Stunden	Sergei Sawitzki
TM035 - Prakt. Reconfigurable Computing, Reconfigurable Co...	Praktikum	Mündliche Prüfung	20 Seiten		4.0	Drittelnoten	Wintersemester	120 Stunden	Sergei Sawitzki

## Lehrinhalte:

- Einführung und Begriffswelt
  - Rekonfigurierbares Rechnen, Rechnerparadigmen
  - ASIC, ASIPS, Mikroprozessoren, FPGAs und ihre funktionale Dichte
  - Einordnung und Klassifizierung der rekonfigurierbaren Systeme
- Schaltungstechnische Basis rekonfigurierbarer Systeme
  - PAL, PLA, PLD
  - CPLD und FPGA
  - hybride Systeme
- Entwurfsfluss und Besonderheiten
  - Hardwareentwurf, Entwurfsschritte
  - Retargierbares Übersetzen
  - Hardware / Software-Codesign
- Anwendungen und Anwendungsentwicklung
  - Klassifizierung
  - Umsetzung
  - Einbindung rekonfigurierbarer Hardware und Kommunikationskonzepte
  - Schnittstellen und Betriebssysteme
- Fortgeschrittene Techniken
  - Dynamische Rekonfiguration
  - Partielle Rekonfiguration
  - Selbstmodifizierende Architekturen
  - System-on-reconfigurable-chip
- Systembeispiele und Fallstudien
  - ISA-orientierte Architekturen
  - Lose gekoppelte Architekturen
  - Datenfluss-Architekturen
- Signale
  - Klassifikation und Analyse
  - Fourier-Transformation
  - Zeit, Frequenz und Bandbreite
- Modulation
  - Formatierung
  - Basisband-Modulation
  - Trägermodulation
  - Impulsformung
- Kanalkodierung
  - Block-Kodes
  - Faltungskodes
  - Iterative Kodierungsverfahren
  - Kodespreizung und -kaskadierung
- Frequenzspreizung und Multiplexverfahren
  - Grundlagen
  - Frequenzspreizung
  - Multiplexverfahren
  - Vielträgermodulation, OFDM-Systeme
- Systemstudien (z. B. wahlweise W-USB, WLAN, DOCSIS oder andere)
- Vorstellung der Aufgabenstellung
- Einarbeitung in die Entwurfswerkzeuge
- Umsetzung und Dokumentation der Aufgabe

## Qualifikationsziele:

Die Studierenden ...



- besitzen eine vertiefte Kenntnis moderner Übertragungssysteme, insbesondere des Aufbaus und der Funktionsweise von Basisband Transceivern
- kennen verschiedene Implementierungsaspekte von digitalen Kommunikationssystemen
- verstehen die Abhängigkeiten zwischen verschiedenen Systemparametern und der erreichbaren Übertragungsqualität
- verstehen Qualitätskriterien digitaler Kommunikationssysteme und Einflussfaktoren bei digitaler Datenübertragung
- sind befähigt digitale Übertragungsstandards zu interpretieren und auf der Ebene der Systemarchitektur (bis hin zur algorithmischen Ebene) zu spezifizieren und zu entwerfen
- kennen rekonfigurierbare Rechnersysteme als Entwurfsvariante des modernen Systementwurfs
- kennen die schaltungstechnische Basis und Hardware-Plattformen des Reconfigurable Computing
- sind befähigt, Vor- und Nachteile einer rekonfigurierbaren Implementierung eines Systems realistisch abschätzen zu können
- können eine Anwendung mit Hilfe rekonfigurierbarer Hardware implementieren

### Verwendbarkeit:

Das Modul "Digitale Kommunikationssysteme und Reconfigurable Computing" baut auf den im Bachelor-Studium erworbenen Kenntnissen und Fähigkeiten auf, wie sie zum Beispiel durch die Module "Einführung in Digitaltechnik", "Rechnerstrukturen und Digitaltechnik", "Informationstechnik", "Übertragungstechnik", "Systemtheorie", "Großintegrierte Systeme" oder ähnliche Module aus den Curricula anderer Hochschulen vermittelt werden und ist daher mit diesen sinnvoll kombinierbar. Die Anwendung bereits erworbener Kompetenzen und Fähigkeiten wird zielgerichtet im Bezug auf moderne Kommunikationssysteme und rekonfigurierbare Rechnerstrukturen weiterentwickelt.

### Voraussetzungen und Empfehlungen:

- grundlegende Kenntnisse analoger und digitaler Signalverarbeitung
- Kenntnisse der Rechnerarchitektur
- Programmierkenntnisse (vorzugsweise C)
- VHDL-Kenntnisse

### Literatur:

- Lüders, Christian: Mobilfunksysteme, Vogel Verlag 2001
- Pehl, Erich: Digitale und analoge Nachrichtenübertragung, Hüthig Verlag 2001
- Werner, Martin: Nachrichtentechnik, Vieweg Verlag 2002
- Read, Richard: Nachrichten und Informationstechnik, Pearson Studium 2004
- Dankmeier, Wilfried: Grundkurs Codierung, Vieweg Verlag 2006
- Tietze, Ulrich; Schenk, Christoph: Halbleiterschaltungstechnik, 15. Auflage, Springer Verlag, 2016
- Sklar, Bernard: Digital Communications. Fundamentals and Applications, 2nd~ edition, Prentice Hall, 2001
- Bobda, Christophe: Introduction to Reconfigurable Computing: Architectures, algorithms and applications, Springer 2007
- Hauck, Scott; DeHon, Andre: Reconfigurable computing: the theory and practice of FPGA-based computation, Morgan Kaufmann Publishers 2008
- Hsiung, Pao-Ann; Santambrogio, Huang, Chun-Hsian: Reconfigurable System Design and Verification, CRC Press 2009

Aufgabenabhängig können weitere anwendungsspezifische Quellen herangezogen werden (z., B. Bildverarbeitung, Kryptographie, digitale Signalverarbeitung usw.)

- Bobda, Christophe: Introduction to Reconfigurable Computing: Architectures, algorithms and applications, Springer 2007
- Hauck, Scott; DeHon, Andre: Reconfigurable computing: the theory and practice of FPGA-based computation, Morgan Kaufmann Publishers 2008
- Hsiung, Pao-Ann; Santambrogio, Huang, Chun-Hsian: Reconfigurable System Design and Verification, CRC Press 2009

### Studiengänge:

- Informatik Master of Science (Wahlmöglichkeit 2. Semester)
- IT Engineering Master of Science (Wahlmöglichkeit 2. Semester)

## ◆ MM044 - Fotorealismus und Simulation

Verantwortliche:	Christian-Arved Bohn
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

### Bestandteile:

Teilleistung	Lernform	Prüfungsform	-umfang	-dauer	ECTS	Benotung	Turnus	Aufwand	Lehrende
TM036 - Fotorealismus und Simulation, Visualisierung	Vorlesung	Mündliche Prüfung		20 Min.	5.0	Drittelnoten	Wintersemester	150 Stunden	Christian-Arved Bohn

### Lehrinhalte:

Über die Grundlagen der Computeranimation hinaus werden in der Veranstaltung fortgeschrittene Algorithmen der Animation und Visualisierung besprochen. Der erste Teil der Veranstaltung behandelt unterschiedliche Modellierungstypen wie Fraktale, Perlin Noise, Lindenmeyersysteme, implizite Modellierung, prozedurale Modellierung und "Articulated Figures". Der zweite Teil erörtert Reaktions-Diffusions-Modelle zur Simulation von Flüssigkeiten und Gasen und zur Modellierung von Stoffen und Haaren - abgerundet durch eine Übersicht über klassische Methoden der Visualisierung von Strömungsvorgängen.

Studierende erhalten einen umfassenden Überblick über aktuelle Techniken der Simulation von Lichtausbreitung, um virtuelle Szenen realitätsnah darzustellen. Der erste Teil der Veranstaltung ist eine Einführung in die Radiometrie. Darauf basierend folgt der grundlegende Algorithmus des Monte Carlo Path Tracing bzw. des Stochastic Path Tracing mit diversen Optimierungsmethoden, wie z. B. Stochastic Ray Tracing versus Stochastic Light Tracing, verschiedene Sampling-Methoden und dem Algorithmus des Photon Mapping.

Mit diesem Wissen über das ideale Lichtmodell werden klassische Methoden aufbereitet (z. B. Radiosity) und deren physikalische Grundlage in der Radiometrie beleuchtet. Der Fokus des zweiten Teils der Veranstaltung ist die Kinetik, d.h. die Bewegung fester Körper unter Einwirkung von Kräften. Die Berechnung dieser bzw. die Rigid Body Simulation wird physikalisch und im Hinblick auf die Verwendung in Computerspielen betrachtet, bei der es darum geht, numerische Probleme der Simulation so zu lösen, dass die Echtzeitberechnung möglich ist. Die Rigid Body Simulation ist Basis für die realistische Bewegung von Körpern virtuellen Szenen.

### Qualifikationsziele:

Studierende erlangen das Verständnis

- über Algorithmen der Visualisierung und
- über fortgeschrittene Algorithmen der Computeranimation.

Studierende erhalten die Fähigkeit zur Einschätzung der Bedeutung der physikalischen Simulation. Verständnis der physikalischen Simulation in der Computergrafik, insbesondere die Simulation von Lichttransfer und Kinetik.

### Verwendbarkeit:

Durch ein nun erreichtes, umfassendes Bild auf sämtliche Themen der Computergrafik ist das Modul der letzte Schritt, um in einem externen Unternehmen Projekte eigenständig bewerten und wesentliche Schritte überwachen zu können. Das Modul ist somit ideal als Vorbereitung auf das "Betriebspraktikum" und ggf. auf die anschließende Bachelor-Arbeit in einem Unternehmen zu sehen.

### Voraussetzungen und Empfehlungen:

Grundlagen der Mathematik, Vektorrechnung, Lineare Algebra, Grundlagen der Computergrafik

### Literatur:

- Henrik W. Jensen: Realistic Image Synthesis Using Photon Mapping, Peters, Wellesley, 2001.
  - Ian Millington: Game Physics Engine Development, Morgan Kaufmann, 2007.
  - Kenny Erleben et al.: Physics-Based Animation, Course Technology, 2005.
- 
- Kenny Erleben et al.: Physics-Based Animation, Course Technology, 2005.

- Alan Watt, Mark Watt: Advanced Animation and Rendering Techniques, Addison Wesley Longman Limited, 1998.
- G. M. Nielson, H. Hagen, H. Müller: Scientific Visualization, IEEE, 1997.
- Charles D. Hansen, Chris R. Johnson: The Visualization Handbook, Academic Press Inc, 2004.

### **Studiengänge:**

- Informatik Master of Science (Wahlmöglichkeit 2. Semester)

## ◆ MM048 - Projekt Informatik

Verantwortliche:	Ulrich Hoffmann
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

### Bestandteile:

Teilleistung	Lernform	Prüfungsform	-umfang	-dauer	ECTS	Benotung	Turnus	Aufwand	Lehrende
TM037 - Projekt Informatik	Projektarbeit	Schriftl. Ausarbeitung (ggf. mit Präsentation)	20 Seiten	30 Min.	5.0	Drittelnoten	jedes Semester	150 Stunden	Ulrich Hoffmann

### Lehrinhalte:

Themenstellungen aus den Arbeitsgruppen und Laboren der Hochschule und aus Kooperationsprojekten mit externen Firmen.

Das Projekt wird in der Hochschule bearbeitet. Die Themenstellungen sollen dabei möglichst interdisziplinär sein, also sowohl Informatik- als auch anwendungsbereichsspezifische Aspekte enthalten.

### Qualifikationsziele:

Selbständiges und eigenverantwortliches Einarbeiten in eine aktuelle Themenstellung aus der Informatik unter zur Hilfenahme aktueller Quellen aus dem wissenschaftlichen Umfeld.

Erkennen der Bedeutung des methodischen und wissenschaftlichen Arbeitens für die Sicherung der Qualität einer Software-Lösung.

Interdisziplinäres Arbeiten und Kommunikation mit Fachleuten aus Informatik-fremden Bereichen.

Praktische Erfahrungen sammeln im Projekt-Management und den Bereichen Projektplanung, Koordination, Aufgabenaufteilung, Zeitmanagement, Delegation und Controlling.

Stärkung der sozialen Kompetenzen in den Bereichen Teamarbeit, Selbstständigkeit, Eigenverantwortung und Selbstorganisation.

### Verwendbarkeit:

Das Modul dient als Vorbereitung auf die Master-Thesis.

### Voraussetzungen und Empfehlungen:

- Vertrautheit mit den grundlegenden Konzepten und Phasen des Projektmanagements (Initiierung, Planung, Durchführung, Abschluss).
- Verständnis der verschiedenen Projektmanagementmethoden (z.B. Wasserfall, Agile).
- Fähigkeit zur Motivation und zum flexiblen Reagieren auf Änderungen.
- Grundlegende Kenntnisse in der Informatik und objektorientierten Programmier Techniken, insbesondere im Kontext von Software-Projekten.

### Literatur:

Themenabhängig

### Studiengänge:

- Informatik Master of Science (2. Semester)

## ◆ MM050 - Master-Thesis

Verantwortliche:	Sergei Sawitzki
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

### Bestandteile:

Teilleistung	Lernform	Prüfungsform	-umfang	-dauer	ECTS	Benotung	Turnus	Aufwand	Lehrende
MTH - Master-Thesis	Thesis	Abschlussarbeit			28.0	Zehntelnoten	jedes Semester	840 Stunden	Sergei Sawitzki

### Lehrinhalte:

themenabhängig

### Qualifikationsziele:

Die Studierenden

- können komplexe Aufgabenstellungen selbständig zu erarbeiten
- können Problemstellungen im größeren Kontext zu verorten
- sind in der Lage wissenschaftliche Methoden für die Problemlösung einzusetzen
- können Ergebnisse überzeugend unter besonderer Berücksichtigung der Aspekte wissenschaftlichen Arbeitens darzustellen

### Verwendbarkeit:

Keine

### Voraussetzungen und Empfehlungen:

Fachliche und persönliche Kompetenzen der zurückliegenden Semester, insbesondere themenabhängig fachverwandte Module

### Literatur:

themenabhängig

### Studiengänge:

- Betriebswirtschaftslehre Master of Science (4. Semester)
- Data Science & Artificial Intelligence Master of Science (3. Semester)
- E-Commerce Master of Science (3. Semester)
- Informatik Master of Science (3. Semester)
- IT-Sicherheit Master of Science (3. Semester)
- IT Engineering Master of Science (3. Semester)
- Sustainable & Digital Business Management Master of Science (3. Semester)
- Wirtschaftsinformatik / IT-Management Master of Science (3. Semester)
- Wirtschaftsingenieurwesen Master of Science (3. Semester)

## ◆ MM058 - Master-Kolloquium

Verantwortliche:	Sergei Sawitzki
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

### Bestandteile:

Teilleistung	Lernform	Prüfungsform	-umfang	-dauer	ECTS	Benotung	Turnus	Aufwand	Lehrende
TM010 - Master-Kolloquium	Kolloquium	Kolloquium		45 Min.	2.0	Drittelnoten	jedes Semester	60 Stunden	Sergei Sawitzki

### Lehrinhalte:

- nach Thema der Master-Arbeit unterschiedlich
- Fachvortrag über Thema der Master-Thesis sowie über die gewählte Vorgehensweise und die Ergebnisse
- Diskussion der Qualität der gewählten Lösung
- Fragen und Diskussion zum Thema der Master-Arbeit und verwandten Gebieten

### Qualifikationsziele:

Die Studierenden ...

- besitzen die Fähigkeit der konzentrierten Darstellung eines intensiv bearbeiteten Fachthemas unter besonderer Berücksichtigung der Aspekte wissenschaftlichen Arbeitens
- verfestigen die Kompetenz, eine fachliche Diskussion über eine Problemlösung und deren Qualität zu führen
- verfügen über ausgeprägte Kommunikations- und Präsentationsfähigkeiten

### Verwendbarkeit:

Keine

### Voraussetzungen und Empfehlungen:

Fachliche und persönliche Kompetenzen der zurückliegenden Semester, insbesondere themenabhängig fachverwandte Module und Master-Thesis

### Literatur:

themenabhängig

### Studiengänge:

- Betriebswirtschaftslehre Master of Science (4. Semester)
- Data Science & Artificial Intelligence Master of Science (3. Semester)
- E-Commerce Master of Science (3. Semester)
- Informatik Master of Science (3. Semester)
- IT-Sicherheit Master of Science (3. Semester)
- IT Engineering Master of Science (3. Semester)
- Sustainable & Digital Business Management Master of Science (3. Semester)
- Wirtschaftsinformatik / IT-Management Master of Science (3. Semester)
- Wirtschaftsingenieurwesen Master of Science (3. Semester)