

Modulhandbuch

Masterstudiengang

»Informatik«

SPO 2017



Verabschiedet am: 02.04.2025

Die Modulbeschreibungen dienen der inhaltlichen Orientierung in Ihrem Studium.

Rechtlich verbindlich ist nur die jeweils geltende Studien- und Prüfungsordnung.

Inhaltsverzeichnis

1	Modul Wissenschaftliche Grundlagen	3
1.1	Angewandte Mathematik	3
1.2	Theoretische Informatik	6
2	Modul Systemarchitekturen	10
2.1	Softwaresysteme	10
2.2	Hardwaresysteme	14
3	Modul Vertiefung	16
3.1	Fachbezogene Wahlpflichtfächer	16
4	Modul Wissenschaftliches Arbeiten	18
4.1	Projektarbeit	18
4.2	Workshop	20
4.3	Master's Seminar	22
5	Modul Masterarbeit	24
5.1	Master Thesis	24

1 Modul Wissenschaftliche Grundlagen

1.1 Angewandte Mathematik

Name / engl.

Angewandte Mathematik / Applied Mathematics

Kürzel

AMATH

Verantwortlicher

Prof. Dr. Stefan Glasauer

Fakultät

Fakultät für Informatik

Lehrsprache

Deutsch

Fakultät für angewandte Geistes- und Naturwissenschaften

Verwendbarkeit

Masterstudiengang Informatik

Dauer / Angebot

ein Semester, jeweils im Wintersemester

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 5,

Präsenzzeit: 60 h, Selbststudium: 90 h, Gesamtaufwand: 150 h

Lehrveranstaltungen

Angewandte Mathematik (4 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Seminar, Übungen, Teilnehmeraktive Lehrveranstaltung

Prüfung

Prüfungsnummer

8900010

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Klausur, 90 Minuten, Hilfsmittel: Taschenrechner,

2 DIN-A4-Seiten handgeschriebene Formelsammlung

Zusätzliche Informationen

hilfreiche Voraussetzungen

Kenntnisse über komplexe Zahlen und lineare Algebra

Inhalte des Moduls

- Diskrete Fourier-Transformation
 - Fourier-Reihen
 - Diskrete Fourier-Transformation (DFT)
 - Schnelle Fourier-Transformation (FFT)
 - Anwendung: Digitale Filter
- Algebraische Strukturen
 - Gruppentheorie
 - Ringe und Körper
 - Zahlentheorie
 - Polynomringe
 - Endliche Körper
 - Anwendung: Public Key-Kryptographie
 - Anwendung: Fehlerkorrigierende Codes

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Die Studierenden erwerben Kenntnisse über algebraische Konzepte und Methoden, die als zeitloses methodisches Rüstzeug für die Beschreibung und Lösung informations- und kommunikationstechnischer Probleme dienen. Die Anwendung dieser Konzepte wird jeweils exemplarisch illustriert in den Bereichen Kryptographie, Codierungstheorie und digitale Filter.

Literaturliste

Burger, Burge: Digitale Bildverarbeitung, Springer 2015

Manz: Fehlerkorrigierende Codes, Springer 2017

Neubauer: Diskrete Fouriertransformation, Vieweg + Teubner 2012

Socher: Algebra für Informatiker, Hanser 2012

Wätjen: Kryptographie, Springer 2018

Witt: Algebraische und zahlentheoretische Grundlagen der Informatik, Springer 2014

1.2 Theoretische Informatik

Name / engl.

Theoretische Informatik / Theoretical Computer Science

Kürzel

THEM

Verantwortlicher

Prof. Dr. Thomas Rist

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Masterstudiengang Informatik

Dauer / Angebot

ein Semester, jeweils im Sommersemester

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 5,

Präsenzzeit: 60 h, Selbststudium: 90 h, Gesamtaufwand: 150 h

Lehrveranstaltungen

Theoretische Informatik (4 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht mit verzahnten Übungseinheiten und aktiver Beteiligung der Teilnehmer*innen.

Prüfung

Prüfungsnummer

8900020

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Klausur, 60 Minuten, keine Hilfsmittel

Zusätzliche Informationen

hilfreiche Voraussetzungen

Grundkenntnisse zu den Themen Formalen Sprachen, endlichen Automaten, Algorithmen und Datenstrukturen, wie sie typischerweise im Grundlagenbereich eines Informatik-Bachelorstudiums vermittelt werden.

Inhalte des Moduls

Theoretische Informatik befasst sich mit den grundsätzlichen Fragestellungen, die bei der Konstruktion und Programmierung von Rechenmaschinen auftreten. Welche Problemstellungen lassen sich überhaupt vernünftig mit einer Rechenmaschine behandeln, welche prinzipiell nicht? Welche Problemstellungen sind inhärent schwierig und bringen - trotz prinzipieller Lösbarkeit - reale Rechenmaschinen schnell an ihre Grenzen? Welche Aussagen über Programme lassen sich automatisch ableiten, welche nicht? Welchen Unterschied macht es, wie solche Maschinen konzipiert sind und welche Ausdrucksmächtigkeit die darauf ausführbaren Programme haben?

Aufbauend auf den im Bachelorstudiengang erworbenen Grundlagenwissen führt der Kurs - den klassischen Weg folgend - kompakt über die Themen „Formale Sprachen“, Grammatiken und Automaten hin zur Turing Maschine und ihren Varianten. Mit der vertieften Diskussion wichtiger Konzepte wie Äquivalenzen zwischen formalen Sprachen und Berechnungsmechanismen, Akzeptanz- und Entscheidbarkeit werden die Teilnehmer*innen an Denkweisen und formale Beweistechniken der Theoretischen Informatik herangeführt und lernen repräsentative Instanzen un- / semi- / entscheidbarer Probleme kennen.

Über eine Auffrischung der aus dem Bachelorstudium als bekannt vorausgesetzten O-Notation zur Einordnung des Ressourcenverbrauchs von Berechnungen, wird der Bogen zur Komplexitätstheorie gespannt. Mit der Einführung wichtiger Komplexitätsklassen und deren Beziehungen zueinander werden die Begriffe Vollständigkeit und Reduktion geklärt. In die Betrachtungen miteinbezogen werden auch neuere Konzeptionen von Rechenmaschinen und Algorithmen (u.a. Parallelrechner, probabilistische Algorithmen, Quantum-Computer).

Anhand formaler Logiksprachen werden die Themen Expressivität, Modell und Kalkül eingeführt und mit Blick auf Berechenbarkeit, Komplexität und deren Bedeutung für andere Teilgebiete der Informatik (u.a. Programmverifikation und symbolische KI / Wissensverarbeitung) diskutiert.

Die im Modul behandelten Themen umfassen:

- Beweistechniken für induktiv definierte Mengen
- Eigenschaften von Sprachklassen der Chomsky-Hierarchie
- Nachweis der Äquivalenzen verschiedener Verarbeitungsmechanismen
- Berechenbarkeit, Konzepte und grundlegende Theoreme (u.a. Halteproblem, PKP, Satz von Church, Satz von Rice)
- Komplexitätstheorie, O-Notation, Komplexitätsklassen, Reduktion
- Komplexität und weitere Komplexitätsklassen im Kontext von Parallelrechner, probabilistische Algorithmen und Quanten-Computer

- Formale Logiksprachen, Definition, Expressivität, Kalküle, Berechenbarkeit, Komplexität und Anwendungsfelder, u.a. Programmverifikation und symbolische Künstliche Intelligenz und Wissensverarbeitung.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Kenntnisse:

Die Studierenden:

- verschaffen sich einen Überblick zu den zentralen Fragestellungen, Herangehensweisen und Erkenntnissen der Theoretischen Informatik sowie deren grundlegende Bedeutung für anwendungsorientierte Informatik,
- kennen für die Gebiete Berechenbarkeit und Komplexität repräsentative Problemstellungen.

Fertigkeiten:

Die Studierenden sind in der Lage:

- grundlegende Argumentationen / Beweisansätze der Theoretischen nachzuvollziehen und an einfachen Problemstellungen selbst durchzuführen,
- Algorithmen und Softwaresysteme hinsichtlich ihrer Komplexität zu analysieren und in passende Komplexitätsklassen einordnen,
- Sachverhalte in einer Logiksprache zu formalisieren

Kompetenzen:

Die Studierenden können:

- erworbenes Wissen und Fähigkeiten auch auf neue Problemstellungen übertragen und anwenden,
- Argumentationen zu Eigenschaften von Algorithmen und Programmen formal untermauern.

Literaturliste

D.W. Hoffmann: Theoretische Informatik. 4. Auflage, Hanser (2018)

R. Socher: Theoretische Grundlagen der Informatik. 2.Auflage, Fachbuchverlag Leipzig (2005)

M. Kreuzer & S. Kühling: Logik für Informatiker. Pearson Studium (2006)

Aktuelle Literaturempfehlungen und Online-Quellen zu behandelten Themen werden im Kurs bekannt gegeben.

2 Modul Systemarchitekturen

2.1 Softwaresysteme

Name / engl.

Softwaresysteme / Software Systems

Kürzel

B.1

Verantwortlicher

Prof. Dr. Philip Heidegger

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Masterstudiengang Informatik

Dauer / Angebot

ein Semester, jeweils im Sommersemester

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 5,

Präsenzzeit: 60 h, Selbststudium: 90 h, Gesamtaufwand: 150 h

Lehrveranstaltungen

Softwaresysteme (4 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Seminar, Laborpraktikum, Studienarbeit, Übung, Teilnehmeraktive Lehrveranstaltung

Prüfung

Prüfungsnummer

8900030

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Studienarbeit, 40 Seiten

Zusätzliche Informationen

hilfreiche Voraussetzungen

Fortgeschrittene Kenntnisse aus den Bereichen Software-Entwicklung und Software-Engineering.

Inhalte des Moduls

Im Rahmen dieser Veranstaltung präsentieren zwei Dozenten Softwaresysteme aus ihren Arbeitsgebieten und diskutieren zentrale Aspekte mit den Studierenden.

1. Systemprogrammierung mit Fokus auf Sicherheit (Prof. Dr.-Ing. Dominik Merli) Systemprogrammiersprachen werden verwendet, um Betriebssysteme, Firmware oder performanceorientierte Applikationen wie Web-Browser zu entwickeln. Die Paradebeispiele C und C++ weisen jedoch seit Jahrzehnten Mängel bzgl. Memory Safety auf, die zu unzähligen Schwachstellen geführt haben. In diesem Kurs wird die Sprache Rust motiviert und eingeführt. Neben der Programmierung in Rust werden auch die Vorteile des umfangreichen Ökosystems diskutiert. Studierende wenden die neu erworbenen Kenntnisse in einem anwendungsnahen Projekt an, um kreative Ideen technisch sauber und sicher umsetzen zu können.

2. Webanwendungen (Prof. Dr. Phillip Heidegger) Viele Anwendungen sollen heutzutage auf einer Vielzahl von Geräten, u.a. auf Handys, PCs, Tablets verwendet werden. Um nicht für jede Plattform eine eigene native Anwendung entwickeln zu müssen, wird häufig auf die Entwicklung von Webanwendungen zurückgegriffen. Da Webanwendungen in Browsern ablaufen, und sich deren Ökosystem stark von dem der traditionellen Betriebssysteme unterscheidet, sind Webanwendungen häufig andersartig strukturiert und Ihre Architektur unterscheidet sich von denen normaler nativer Anwendungen. Nach einer kurzen Einführung in die Architektur traditioneller GUI basierter Anwendungen werden die Unterschiede und die damit verbundenen Herausforderungen von Webanwendungen systematisch analysiert und mögliche Lösungen evaluiert. Wichtige Technologien, die vorgestellt werden, sind TypeScript, JQuery und Angular.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Nach erfolgreicher Teilnahme am Modul sind die Studierenden in der Lage:

- Besonderheiten der für die vorgestellten Softwaresysteme verwendeten Technologien zu verstehen.
- Einschlägige Programmiertechniken für die Erstellung von Softwaresystemen anzuwenden.
- Anforderungen und Herausforderungen an Softwaresysteme systematisch zu analysieren.
- Vorgegebene Beispiele systematisch bezüglich Effektivität und Effektivität zu beurteilen.
- Softwaresysteme aus existierenden Komponenten zu konstruieren.

Literaturliste

W. Burger, M. J. Burge: Digitale Bildverarbeitung, Springer (2015)

Pacheco, P.: Introduction to Parallel Programming, Morgan Kaufman, 2011

I. Ristic: Bulletproof SSL and TLS, Feisty Duck (2015)

Insight Segmentation and Registration Toolkit (ITK),

www.itk.org

<https://angular.io>

<https://www.typescriptlang.org>

2.2 Hardwaresysteme

Name / engl.

Hardwaresysteme / Hardware Systems

Kürzel

HWS

Verantwortlicher

Prof. Dr. Gundolf Kiefer

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Masterstudiengang Informatik

Dauer / Angebot

ein Semester, jeweils im Sommersemester

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 5,

Präsenzzeit: 60 h, Selbststudium: 90 h, Gesamtaufwand: 150 h

Lehrveranstaltungen

Hardwaresysteme (2 SWS)

Praktikum Hardwaresysteme (2 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Laborpraktikum zur Anwendung und Vertiefung der erworbenen Kenntnisse, Teilnehmeraktive Lehrveranstaltung

Prüfung

Prüfungsnummer

8900040

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Klausur, 60 Minuten, keine Hilfsmittel

Zusätzliche Informationen

hilfreiche Voraussetzungen

Keine

Inhalte des Moduls

- Technologische Grundlagen & Trends
- Modellierungssprachen (z.B. VHDL oder SystemC)
- Spezialisierte Rechnerarchitekturen (z.B. für Grafik- oder KI-Anwendungen)
- Systementwicklung mit FPGA-Technologie
- Hardware-orientierte Software-Entwicklung

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Die Studierenden sind in der Lage,

- digitale Systeme mit FPGAs in VHDL oder SystemC zu entwickeln.
- die Einsatzmöglichkeiten von Standard-Prozessoren, Spezialprozessoren sowie FPGA-Technologie einzuordnen.
- die Effizienz alternativer Hardware-Architekturen zu beurteilen.
- Hardware-Systemarchitekturen zu bewerten.

Literaturliste

Die Literatur wird zu Semesterbeginn in der Vorlesung bekannt gegeben.

3 Modul Vertiefung

3.1 Fachbezogene Wahlpflichtfächer

Name / engl.

Fachbezogene Wahlpflichtfächer / Optional required Subjects

Kürzel	Verantwortlicher Professorinnen und Professoren der Fakultät für Informatik
Lehrsprache Siehe Angaben des jeweiligen FWP-Fachs	Fakultät Fakultät für Informatik
Verwendbarkeit Masterstudiengang Informatik	Dauer / Angebot Die Dauer des Moduls beträgt ein Semester. Das Modul wird sowohl im Sommersemester als auch im Wintersemester angeboten.
Arbeitsaufwand / Zusammensetzung SWS: 10 - 20 (*), ECTS: 25 Präsenzzeit: 112,5 h - 225 h Selbststudium: 637,5 h - 525 h Gesamtaufwand: 750 h (*) Die Anzahl der SWS des Vertiefungsmoduls kann je nach Art der gewählten Lehrveranstaltungen zwischen 10 und 20 SWS betragen.	
Lehrveranstaltungen Die FWP-Fächer können aus dem Angebot der Fakultät für Informatik ausgewählt werden.	
Lehr-/Lernmethoden Directed Reading, Seminaristischer Unterricht, Seminar, Übungen, Praktikum, Teilnehmeraktive Lehrveranstaltung	
Prüfung	
Prüfungsnummer	Benotung Gemäß § 20 der APO in der jeweils gültigen Fassung.
Prüfungsform Weitere Informationen zu den fachbezogenen Wahlpflichtfächern finden Sie auf der Webseite des Studiengangs unter Studienrelevante Downloads.	

Zusätzliche Informationen

hilfreiche Voraussetzungen

Informationen zur Teilnahme an Wahlpflichtfächern können der Studien- und Prüfungsordnung entnommen werden.

Inhalte des Moduls

Spezifische Fachkompetenz in den einzelnen Fächern.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Spezifische Fachkompetenz in den jeweiligen Fächern.

Gewichtung der Einzelleistung in der Gesamtnote

Weitere Informationen zu den fachbezogenen Wahlpflichtfächern finden Sie auf der Webseite des Studiengangs unter Studienrelevante Downloads.

Literaturliste

Literatur wird im Seminar bekannt gegeben.

4 Modul Wissenschaftliches Arbeiten

4.1 Projektarbeit

Name / engl.

Projektarbeit / Project Work

Kürzel

PA

Verantwortlicher

Studiengangsleiter

Lehrsprache

Das Modul wird in deutscher Sprache unterrichtet.

Fakultät

Fakultät für Informatik

Verwendbarkeit

Masterstudiengang Informatik

Dauer / Angebot

Das Modul wird regelmäßig im Sommersemester angeboten und hat eine Laufzeit von zwei Semestern.

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 10,

Präsenzzeit: 60 h, Selbststudium: 240 h, Gesamtaufwand: 300 h

Lehrveranstaltungen

Projektarbeit (4 SWS)

Lehr-/Lernmethoden

Studierende erarbeiten in Kleingruppen weitestgehend eigenständig Lösungen zu einem praxisorientierten Thema aus dem IT-Bereich.

Ziel ist es einen Projektablauf möglichst realitätsnah mit allen Facetten abzubilden. Die Projektthemen werden von Prüfungsberechtigten der Fakultät für Informatik vergeben und umfassen einen praktischen Teil (Software/Hardware), eine Dokumentation (Studienarbeit) und eine Präsentation.

Die Präsentation findet in der Regel im Rahmen eines Projekttagess statt. Die Abstimmung mit dem Projektsteller erfolgt in regelmäßigen persönlichen Treffen und über elektronische Kanäle. Die Bearbeitung ist nicht notwendigerweise an die Vorlesungszeit gebunden.

Prüfung

Prüfungsnummer

8900090

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Portfolioprüfung:

- Projektarbeit, 20-40 Seiten, 80%
 - Präsentation, 20-40 Minuten, 20%
-

Zusätzliche Informationen

hilfreiche Voraussetzungen

Solide Kenntnisse aus den wichtigsten Themenbereichen der Informatik, wie z.B. Algorithmen und Datenstrukturen, Programmieren, Datenbanken, Datenkommunikation, Rechnerstrukturen und Betriebssysteme.

Inhalte des Moduls

Die Studierenden führen in kleinen Teams IT-Projekte durch. Zu den Aufgaben der Studierenden zählen Projektmanagement, sowie je nach Projekt die klassische oder agile Softwareentwicklung, Dokumentation und Präsentation der Ergebnisse.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Nach erfolgreicher Teilnahme am Modul sind die Studierenden in der Lage:

- Teamprozesse zu verstehen und typische Teamkonflikte zu lösen.
- Softwareprojekte im Team hinsichtlich Zeit, Aufwände und Ressourcen eigenverantwortlich zu planen und durchzuführen.
- Agile oder klassische Softwareentwicklungsmethoden praktisch anzuwenden.
- Selbstständiges Erlernen von Methoden und Verfahren.
- Analysieren und Bewerten von Methoden im Hinblick auf die Projektziele.
- Projektergebnisse verständlich zu dokumentieren und ansprechend zu präsentieren.

Literaturliste

Projektspezifische Literatur wird vom Betreuer vor Beginn des Projektes bekanntgegeben.

4.2 Workshop

Name

Workshop / Workshop

Code

WSH

Coordinator

Prof. Dr.-Phil. Alessandra Zarcone

Teaching language

English

Faculty

Faculty of Computer Science

Usage possibilities

Masterstudiengang Informatik

Duration / Frequency

1 semester, winter semester, summer semester

Total workload and its constituent parts

Credit hours: 2, CP credits: 5,

Contact hours: 30h, Independent study: 120h, Total workload: 150h

Courses

Workshop (2 credit hours)

Teaching and learning methods

Seminar format with practical exercises

Exam

Grading

Examination number

8900100

According to § 20 of the APO in the currently valid version.

Type of exam / required course achievements

Portfolio exam:

- Written assignment, 3 - 5 pages, 60%
- Oral examination, 20 minutes, 40%

Additional Information

Prerequisites

None

Usage possibilities

Master's degree Computer Science

Content of the module

We will cover different aspects and tools of a researcher's everyday work, in particular:

- Research methods and evaluation of statistical models
- Documentation of data collection
- Ethical aspects
- Scientific Writing
- Bibliographic search and citing the sources
- Presentation of scientific writing (talks and posters)
- Critical approach to one's own scientific work and to the literature and peer review
- Networking and promoting your work

At the beginning of the course the students are assigned one paper each. The papers are announced in advance so that every student who is interested in the module can sign up online and select a paper. The students are required to critically analyze the paper regarding the aspects covered by the course, to compile a review of the paper regarding these aspects and to prepare a poster presenting the content of the paper.

At the end of the course the students present the papers in a poster session and are required to actively participate in the discussion of at least two more posters.

We strongly recommend attending the Workshop at the beginning of your Master (1st or 2nd semester).

Qualification aims for the module learning objectives/skills

After successfully completing this module, the students are able to:

- Critically review existing research work
- Plan their own research work
- Carry out their own research by adopting standard practice methods
- Present their own work in an accurate and effective way

Reading list

Booth, Wayne C, et al. The Craft of Research: University of Chicago Press, 4. edition, (2016)

More material (videos, papers) will be provided during the course.

Software:

- LaTeX: <https://www.latex-project.org>

4.3 Master's Seminar

Name

Masterseminar / Master's Seminar

Code

MASEM

Coordinator

Prof. Dr. Phil. Alessandra Zarcone

Teaching language

English

Faculty

Faculty of Computer Science

Usage possibilities

Masterstudiengang Informatik

Duration / Frequency

1 semester, winter semester, summer semester

Total workload and its constituent parts

Credit hours: 2, CP credits: 5,

Contact hours: 30h, Independent study: 120h, Total workload: 150h

Courses

Master's seminar (2 credit hours)

Teaching and learning methods

The topics are announced in advance so that every student who is interested in the module can sign up online, select a topic and do their own research.

The students are required to submit an exposé in the first part of the semester and to give a presentation in one of the planned slots. Each presentation is followed by a discussion where the students discuss possible open questions. A critical but productive approach towards one's own work and the peers' work is encouraged.

The students submit an article summarizing their results at the end of the course.

Exam

Examination number

8900110

Grading

According to § 20 of the APO in the currently valid version.

Type of exam / required course achievements

Portfolio exam:

- Written assignment, 10 - 20 pages, 70%
- Oral examination, 20 minutes, 30%

Additional Information

Prerequisites

Module Workshop (WSH)

Usage possibilities

Master's degree Computer Science

Content of the module

The aim of the master's seminar is to promote autonomous research work on a particular topic.

The students expand their knowledge and understanding of the chosen topic and investigate the methodology, potential critical points and open questions. They summarise the results in a long abstract and then present the results to the class. The students review their peers's work and adopt a critical approach towards their own work. The content of all presentations is thematically related to encourage discussion. Each student submits a paper on the chosen topic.

We strongly recommend attending the Workshop "Introduction to Scientific Research" before the Masterseminar.

Qualification aims for the module learning objectives/skills

The students have refined their critical insights and have developed their scientific writing and presentation skills.

They are able to

- research a chosen topic
- structure and write a scientific publication
- apply common-practice scientific methodologies
- analyze and discuss research results
- identify weak points and strength of scientific publications and provide feedback to their peers

Reading list

5 Modul Masterarbeit

5.1 Master Thesis

Name / engl.

Master Thesis / Master Thesis

Kürzel

M

Verantwortlicher

Professorinnen und Professoren der Fakultät für Informatik

Lehrsprache

Nach Absprache mit dem Betreuer

Fakultät

Fakultät für Informatik

Verwendbarkeit

Masterstudiengang Informatik

Dauer / Angebot

Das Modul wird sowohl im Sommersemester als auch im Wintersemester angeboten.

Arbeitsaufwand / Zusammensetzung

ECTS: 25, Arbeitsstunden: 750

Lehrveranstaltungen

Masterarbeit

Lehr-/Lernmethoden

Wissenschaftliches Arbeiten

Prüfung

Prüfungsnummer

9050

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Portfolioprüfung:

- Studienarbeit, 30-100 Seiten, 80%
- Präsentation, 20-30 Minuten, 20%

Zusätzliche Informationen

hilfreiche Voraussetzungen

Informationen zur Masterarbeit können unter § 9 der Studien- und Prüfungsordnung entnommen werden.

Inhalte des Moduls

Das Studium beinhaltet eine Abschlussarbeit (Masterarbeit). Die Masterarbeit soll zeigen, dass die Studierenden in der Lage sind ein Problem aus dem Bereich der Informatik selbständig auf wissenschaftlicher Grundlage zu bearbeiten.

Das Thema der Abschlussarbeit wird vom Studierenden eigenständig gesucht. Im Interesse einer raschen Praxiseingliederung der Studierenden soll die Masterarbeit vorwiegend im Rahmen eines Projekts mit einem Partner aus Industrie, Wirtschaft oder Verwaltung angefertigt werden. Natürlich können auch hochschulinterne Themen in Absprache mit dem Betreuenden ausgewählt werden.

Das Thema der Masterarbeit soll so beschaffen sein, dass sie bei zusammenhängender ausschließlicher Bearbeitung in der Regel in 6 Monaten abgeschlossen werden kann. Die Frist von der Ausgabe bis zur Abgabe der Arbeit beträgt 6 Monate mit der Möglichkeit der einmaligen Verlängerung bis zu vier Monaten bei Glaubhaftmachung nicht selbst zu vertretender Verzögerungsgründe.

Die Anforderungen an die Darstellung des Standes der wissenschaftlichen Erkenntnisse und an die Originalität/Neuartigkeit der Problemlösung liegen deutlich über denen an eine Bachelorarbeit (was sich auch im zeitlichen Aufwand ausdrückt).

Durch die Masterarbeit sollen Kompetenzen auf den folgenden Gebieten nachgewiesen werden:

- Themenspezifisches Fachwissen aus dem Umfeld der Wirtschaftsinformatik,
- Techniken, Methoden und Vorgehensweisen des wissenschaftlichen Arbeitens,
- Projektmanagement bzgl. Zeit und Scope

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Nach erfolgreicher Teilnahme am Modul sind die Studierenden in der Lage:

- ein Problem selbständig unter Verwendung wissenschaftlicher Techniken, Methoden und Vorgehensweisen eigenständig zu lösen
- unterschiedliche Lösungsansätze zu identifizieren, zu bewerten, anzuwenden und in nachvollziehbarer und überzeugender Weise darzustellen
- umfangreiche Aufgaben- und Problemstellungen eigenständig zu strukturieren, zu planen und in begrenzter Zeit und mit begrenztem Budget abzuschließen.

Literaturliste

Theisen, R. T.: Wissenschaftliches Arbeiten. Erfolgreich bei Bachelor- und Masterarbeit, Vahlen 2017

Lück, W., Henke, M.: Technik des wissenschaftlichen Arbeitens. Seminararbeit, Diplomarbeit, Dissertation, Oldenbourg 2009

Rössl, Dietmar: Die Diplomarbeit in der Betriebswirtschaftslehre, facultas.wuv 2008

Eid, M., Gollwitzer, M., Schmitt, M.: Statistik und Forschungsmethoden, Beltz 2015

Index

Angewandte Mathematik , 3

Fachbezogene Wahlpflichtfächer , 16

Hardwaresysteme , 14

Master Thesis , 24

Master's Seminar , 22

Projektarbeit , 18

Softwaresysteme , 10

Theoretische Informatik , 6

Workshop , 20