

Modulhandbuch
Master-Studiengang
Informatik

Prüfungsordnung 20.0

Wedel, den 16. Dezember 2021

Teil I

Modulhandbuch

Kapitel 1.1

Modulhandbuch

Modulverzeichnis nach Modulkürzel

| | | |
|------|---|----|
| M003 | Algorithmics | 13 |
| M005 | Funktionale Programmierung | 17 |
| M006 | Learning and Softcomputing | 21 |
| M009 | Workshop Cryptography | 27 |
| M010 | Aktuelle Entwicklungen in der Informatik | 69 |
| M018 | Robotics | 32 |
| M023 | Seminar Informatik | 25 |
| M027 | Konzepte der Datenbanktechnologie | 46 |
| M029 | Berechenbarkeit und Verifikation | 41 |
| M033 | Methoden der Künstlichen Intelligenz | 54 |
| M035 | Distributed Systems | 50 |
| M042 | Digitale Kommunikationssysteme und Reconfigurable Computing | 59 |
| M044 | Fotorealismus und Simulation | 66 |
| M048 | Projekt Informatik | 57 |
| M050 | Master-Thesis | 71 |
| M058 | Master-Kolloquium | 73 |
| M062 | Praktikum Virtuelle Realität und Simulation | 36 |
| M162 | Moderne Software-Architekturen | 38 |

Modulverzeichnis nach Modulbezeichnung

| | |
|---|----|
| Aktuelle Entwicklungen in der Informatik | 69 |
| Algorithmics | 13 |
| Berechenbarkeit und Verifikation | 41 |
| Digitale Kommunikationssysteme und Reconfigurable Computing | 59 |
| Distributed Systems | 50 |
| Fotorealismus und Simulation | 66 |
| Funktionale Programmierung | 17 |
| Konzepte der Datenbanktechnologie | 46 |
| Learning and Softcomputing | 21 |
| Master-Kolloquium | 73 |
| Master-Thesis | 71 |
| Methoden der Künstlichen Intelligenz | 54 |
| Moderne Software-Architekturen | 38 |
| Praktikum Virtuelle Realität und Simulation | 36 |
| Projekt Informatik | 57 |
| Robotics | 32 |
| Seminar Informatik | 25 |
| Workshop Cryptography | 27 |

I.1.1 Erläuterungen zu den Modulbeschreibungen

Im Folgenden wird jedes Modul in tabellarischer Form beschrieben. Die Reihenfolge der Beschreibungen richtet sich nach der Abfolge im Curriculum.

Vor den Modulbeschreibungen sind zwei Verzeichnisse aufgeführt, die den direkten Zugriff auf einzelne Modulbeschreibungen unterstützen sollen. Ein Verzeichnis listet die Modulbeschreibungen nach Kürzel sortiert auf, das zweite Verzeichnis ist nach Modulbezeichnung alphabetisch sortiert.

Die folgenden Erläuterungen sollen die Interpretation der Angaben in einzelnen Tabellenfeldern erleichtern, indem sie die Annahmen darstellen, die beim Ausfüllen der Felder zugrunde gelegt wurden.

Angaben zum Modul

| | |
|--------------------------|---|
| Modulkürzel: | FH-internes, bezogen auf den Studiengang eindeutiges Kürzel des Moduls |
| Modulbezeichnung: | Textuelle Kennzeichnung des Moduls |
| Lehrveranstaltungen: | Lehrveranstaltungen, die im Modul zusammen gefasst sind, mit dem FH-internen Kürzel der jeweiligen Leistung und ihrer Bezeichnung |
| Prüfung im Semester: | Auflistung der Semester, in denen nach Studienordnung erstmals Modulleistungen erbracht werden können |
| Modulverantwortliche(r): | Die strategischen Aufgaben des Modulverantwortlichen umfassen insbesondere: <ul style="list-style-type: none">▪ Synergetische Verwendung des Moduls auch in weiteren Studiengängen▪ Entwicklung von Anstößen zur Weiterentwicklung der Moduls und seiner Bestandteile▪ Qualitätsmanagement im Rahmen des Moduls (z. B. Relevanz, ECTS-Angemessenheit)▪ Inhaltsübergreifende Prüfungstechnik. Die operativen Aufgaben des Modulverantwortlichen umfassen insbesondere: <ul style="list-style-type: none">▪ Koordination von Terminen in Vorlesungs- und Klausurplan▪ Aufbau und Aktualisierung der Modul- und Vorlesungsbeschreibungen▪ Zusammenführung der Klausurbestandteile, die Abwicklung der Klausur (inkl. Korrekturüberwachung bis hin zum Noteneintrag) in enger Zusammenarbeit mit den Lehrenden der Modulbestandteile |

- Funktion als Ansprechpartner für Studierende des Moduls bei sämtlichen modulbezogenen Fragestellungen.

| | |
|------------------------------|--|
| Zuordnung zum Curriculum: | Auflistung aller Studiengänge, in denen das Modul auftritt |
| Querweise: | Angabe, in welchem Zusammenhang das Modul zu anderen Modulen steht |
| SWS des Moduls: | Summe der SWS, die in allen Lehrveranstaltungen des Moduls anfallen |
| ECTS des Moduls: | Summe der ECTS-Punkte, die in allen Lehrveranstaltungen des Moduls erzielt werden können |
| Arbeitsaufwand: | Der Gesamtarbeitsaufwand in Stunden ergibt sich aus den ECTS-Punkten multipliziert mit 30 (Stunden). Der Zeitaufwand für das Eigenstudium ergibt sich, wenn vom Gesamtaufwand die Präsenzzeiten abgezogen werden. Diese ergeben sich wiederum aus den Semesterwochenstunden (SWS), die multipliziert mit 45 (Minuten) geteilt durch 60 die Präsenzzeit ergeben. |
| Voraussetzungen: | Module und Lehrveranstaltungen, die eine inhaltliche Grundlage für das jeweilige Modul darstellen. Bei Lehrveranstaltungen ist der Hinweis auf das jeweilige Modul enthalten, in dem die Lehrveranstaltung als Bestandteil auftritt. |
| Dauer: | Anzahl der Semester die benötigt werden, um das Modul abzuschließen |
| Häufigkeit: | Angabe, wie häufig ein Modul pro Studienjahr angeboten wird (jedes Semester bzw. jährlich) |
| Studien-/Prüfungsleistungen: | Auflistung aller Formen von Leistungsermittlung, die in den Veranstaltungen des Moduls auftreten |
| Sprache: | In der Regel werden die Lehrveranstaltungen aller Module auf Deutsch angeboten. Um Gaststudierenden unserer Partnerhochschulen, die nicht der deutschen Sprache mächtig sind, die Teilnahme an ausgewählten Lehrveranstaltungen zu ermöglichen, ist die Sprache in einigen Modulen als "deutsch/englisch" deklariert. Dieses wird den Partnerhochschulen mitgeteilt, damit sich die Interessenten für ihr Gastsemester entsprechende Veranstaltungen herausuchen können. |
| Lernziele des: | Übergeordnete Zielsetzungen hinsichtlich der durch das Modul zu vermittelnden Kompetenzen und Fähigkeiten aggregierter Form |

Angaben zu den Lehrveranstaltungen

| | |
|------------------------|---|
| Lehrveranstaltung: | Bezeichnung der Lehrveranstaltung, die im Modul enthalten ist |
| Dozent(en): | Namen der Dozenten, die die Lehrveranstaltung durchführen |
| Hörtermin: | Angabe des Semesters, in dem die Veranstaltung nach Studienordnung gehört werden sollte |
| Art: | Angabe, ob es sich um eine Pflicht- oder Wahlveranstaltung handelt |
| Lehrform: | Lehrform kann Vorlesung, Praktikum, Seminar, u.v.m. sein |
| Semesterwochenstunden: | Eine Semesterwochenstunde dauert 70 Minuten und entspricht einer Vorlesungseinheit |
| ECTS: | Angabe der ECTS-Punkte, die in dieser Lehrveranstaltung des Moduls erzielt werden können |
| Medienformen: | Auflistung der Medienform(en), die in der Veranstaltung eingesetzt werden |
| Lernziele: | Stichwortartige Nennung die zentralen Lernziele der Lehrveranstaltung |
| Inhalt: | Gliederungsartige Auflistung der wesentlichen Inhalte der Lehrveranstaltung |
| Literatur: | Auflistung der wesentlichen Quellen, die den Studierenden zur Vertiefung zu den Veranstaltungsinhalten empfohlen werden. Es wird keine vollständige Auflistung aller Quellen gegeben, die als Grundlage für die Veranstaltung dienen. |

I.1.2 Algorithmics

M003 Algorithmics

| | |
|---------------------------------|--|
| Studiengang | Master-Studiengang Informatik |
| Kürzel | M003 |
| Bezeichnung | Algorithmics |
| Lehrveranstaltung(en) | M003a Algorithmics |
| Verantwortliche(r) | Prof. Dr. Sebastian Iwanowski |
| Zuordnung zum Curriculum | Data Science & Artificial Intelligence (Master) IT Engineering (Master) IT-Sicherheit (Master) Informatik (Master) |
| 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. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | <ul style="list-style-type: none">▪ Verstehen grundlegender mathematischer Konzepte wie Definitionen, Sätze und Beweise.▪ Fähigkeit zur logisch fundierten Formulierung von Sachverhalten.▪ Die Studenten müssen in der Lage sein, die Beweise von Beginn des Kurses nachzuvollziehen.▪ Erforderlich sind ausgezeichnete Kenntnisse der Grundlagen der diskreten Mathematik, insbesondere der Zahlentheorie und der Graphentheorie.▪ Die Studenten müssen über gute Programmierkenntnisse und Erfahrung in der Implementierung grundlegender Algorithmen verfügen. |
| Dauer | 1 |

Lernziele

Die Studierenden wissen, wie man die Effizienz von Algorithmen mit theoretisch fundierten Methoden bewerten kann. Für ausgewählte Anwendungsbereiche wissen sie, wie man deren

Algorithmen detailliert beschreibt, Beispiele dazu konstruiert und implementiert. Sie sind in der Lage, grundlegende Beweise für Effizienz und Korrektheit selbstständig durchzuführen. Sie können auch komplizierte Beweise verstehen und sie anderen Personen erklären.

I.1.2.1 Algorithmics

| | |
|---------------------------------|---|
| Lehrveranstaltung | Algorithmics |
| Dozent(en) | Sebastian Iwanowski |
| Hörtermin | 2 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung mit integrierter Übung/Workshop/Assigm. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Prüfungsform | Klausur / Mündliche Prüfung |
| Sprache | english |
| Lehr- und Medienform(en) | Handout, Overheadfolien, Softwaredemonstration, Tafel |

Lernziele

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.

Inhalt

- 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)

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

I.1.3 Funktionale Programmierung

M005 Funktionale Programmierung

| | |
|---------------------------------|--|
| Studiengang | Master-Studiengang Informatik |
| Kürzel | M005 |
| Bezeichnung | Funktionale Programmierung |
| Lehrveranstaltung(en) | M005a Funktionale Programmierung M005b Übg. Funktionale Programmierung |
| Verantwortliche(r) | Priv.-Doz. Dr. Frank Huch |
| Zuordnung zum Curriculum | IT-Sicherheit (Master) Informatik (Master) |
| Verwendbarkeit | Das Modul kann sinnvoll im Modul "Methoden der Künstlichen Intelligenz", in Projekten und der Master-Thesis genutzt werden. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | 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. |
| Dauer | 1 |

Lernziele

In diesem Modul werden fortgeschrittenen Techniken der funktionalen Programmierung am Beispiel der Sprache Haskell behandelt. Hierzu gehören der Umgang mit Funktionen höherer Ordnung, das Arbeiten mit generischen Datentypen und mit Typklassen, und mit Monaden und Arrows. Es werden beispielhaft eingebettete problemspezifische Sprachen (EDSL) vorgestellt. Dieses Modul soll außerdem die Abstraktion, die Modellbildung stärken und das aus der Mathematik bekannte präzise Arbeiten auf die Software-Entwicklung übertragen. Die Studierenden erfassen, warum funktionale Programmierung gegenüber anderen Programmieransätzen die IT-Sicherheit erhöhen kann, da sie typische Angriffslücken grundsätzlich vermeidet. Die Studierenden lernen, warum Kernelemente funktionaler Programmierung, insbesondere die Seiteneffektfreiheit und die starke Typisierung, besonders geeignet sind, Sicherheitsaspekte von Software zu gewährleisten und nachzuweisen.

I.1.3.1 Funktionale Programmierung

| | |
|---------------------------------|-----------------------------|
| Lehrveranstaltung | Funktionale Programmierung |
| Dozent(en) | Uwe Schmidt |
| Hörtermin | 2 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 2 |
| ECTS | 2.0 |
| Prüfungsform | Klausur / Mündliche Prüfung |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Softwaredemonstration |

Lernziele

Die Studierenden ...

- lernen fortgeschrittene Techniken der funktionalen Programmierung am Beispiel der Sprache Haskell kennen.
- können mit Funktionen höherer Ordnung, mit generischen Datentypen und Typklassen, mit Funktoren, Monaden, Monoiden und weiteren mathematischen Strukturen umgehen.
- lernen die Software-Realisierung mit eingebetteten problemspezifischen Sprachen kennen.
- stärken die Fähigkeiten in der Modellbildung und Abstraktion.
- lernen die Bezüge zwischen Mathematik und funktionaler Programmierung kennen.
- kennen die Vor- und Nachteile des funktionalen Paradigmas für Anwendungen der IT-Sicherheit.

Inhalt

- Einleitung
 - Grundlegende Konzepte
 - Syntax von Haskell
- Datentypen
 - Einfache Datentypen
 - Produkt- und Summen-Datentypen
 - Listen
 - Funktionen höherer Ordnung für Listen
- Typcheck
- Korrektheitsargumentationen
- Rekursive Datenstrukturen

- Bäume
- Bedarfsauswertung
 - Unendliche Strukturen
- Funktoren und Monaden
 - Maybe- und Listen-Monade
 - Zustands-Monade und Ein- und Ausgabe
 - weitere Varianten von Monaden
- Fallstudien
 - Eingebettete problemspezifische Sprachen
 - Monadische Parser
- Parallele und nebenläufige Programmierung
- Testen

Literatur

- Uwe Schmidt:
Funktionale Programmierung,
Vorlesungsunterlagen im Web: <http://www.fh-wedel.de/si/vorlesungen/fp/fp.html>
- Bird, Richard:
Introduction to Functional Programming using Haskell,
2nd Edition Prentice Hall, New Jersey, 1998, ISBN: 0-13-484346-0
- Graham Hutton: Programming in Haskell, Cambridge University Press, 2007, ISBN: 978-0-521-69269-4

I.1.3.2 Übg. Funktionale Programmierung

| | |
|---------------------------------|---------------------------------|
| Lehrveranstaltung | Übg. Funktionale Programmierung |
| Dozent(en) | Uwe Schmidt |
| Hörtermin | 2 |
| Häufigkeit | jährlich |
| Lehrform | Übung/Praktikum/Planspiel |
| Semesterwochenstunden | 2 |
| ECTS | 3.0 |
| Prüfungsform | Abnahme |
| Sprache | deutsch |
| Lehr- und Medienform(en) | studentische Arbeit am Rechner |

Lernziele

Ziel der Übung ist das Erlernen des praktischen Anwenden der Methoden und Konzepte aus der Vorlesung.

Inhalt

Praktische Übungen über die Themen

- Rekursion,
- Typisierung,
- Listen und Tuple,
- Funktionen als Daten,
- Funktoren und Monaden,
- Ein-und Ausgabe.

Literatur

- Uwe Schmidt:
Funktionale Programmierung,
Vorlesungsunterlagen im Web: <http://www.fh-wedel.de/si/vorlesungen/fp/fp.html>
- Bird, Richard:
Introduction to Functional Programming using Haskell,
2nd Edition Prentice Hall, New Jersey, 1998, ISBN: 0-13-484346-0
- Graham Hutton: Programming in Haskell, Cambridge University Press, 2007, ISBN: 978-0-521-69269-4

I.1.4 Learning and Softcomputing

M006 Learning and Softcomputing

| | |
|---------------------------------|---|
| Studiengang | Master-Studiengang Informatik |
| Kürzel | M006 |
| Bezeichnung | Learning and Softcomputing |
| Lehrveranstaltung(en) | M006a Learning & Softcomputing |
| Verantwortliche(r) | Prof. Dr. Ulrich Hoffmann |
| Zuordnung zum Curriculum | Data Science & Artificial Intelligence (Master) IT-Sicherheit (Master) Informatik (Master) Wirtschaftsinformatik/IT-Management (Master) |
| 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. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Voraussetzungen dieses Moduls sind Kenntnisse und praktische Erfahrungen in höheren Programmiersprachen. Außerdem werden mathematische Grundkenntnisse und Kenntnisse der Stochastik erwartet. |
| Dauer | 1 |

Lernziele

Studierende erwerben Kenntnisse im Bereich des maschinellen Lernens. Sie beherrschen die wesentlichen Techniken, mit deren Hilfe Computersysteme Klassifizierungen und Bewertungen durchführen, und sie können sie nach Einsatzgebiet und Güte bewerten und beurteilen. Sie kennen die Herausforderungen die beim Parametrieren von überwachtem Lernenverfahren bedeutsam sind und können sie praktisch anwenden. Sie sind mit wesentlichen Funktionalitäten gängiger Machine-Learning-Bibliotheken vertraut. Sie sind in der Lage eigenständig Aufgaben des maschinellen Lernens zu analysieren, geeignete Methoden auszuwählen und umzusetzen.

Im praktischen Teil erwerben sie zusätzlich die Kompetenz arbeitsteilig in einer kleinen Arbeitsgruppe wissenschaftlich, selbständig an einer umfangreichen Aufgabe Kenntnisse zusammenzutragen und Lösungen zu erarbeiten sowie diese verständlich und strukturiert zu präsentieren. Darüber hinaus erwerben oder vertiefen sie ihr Fachwissen über das Thema der zu bearbeitenden Aufgabe des maschinellen Lernens, z.B. Bildverarbeitung, IT-Sicherheit, E-Commerce oder Betriebswirtschaftslehre.

I.1.4.1 Learning & Softcomputing

| | |
|---------------------------------|-----------------------------|
| Lehrveranstaltung | Learning & Softcomputing |
| Dozent(en) | Ulrich Hoffmann |
| Hörtermin | 2 |
| Häufigkeit | jährlich |
| Lehrform | mehrere Veranstaltungsarten |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Prüfungsform | Assessment |
| Sprache | deutsch |
| Lehr- und Medienform(en) | |

Lernziele

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 Sachverhalts auseinandersetzen.
- besitzen vertiefte Kenntnisse des Themas der konkret bearbeiteten Machine-Learning-Aufgabe, also etwa zu Bildverarbeitung, IT-Sicherheit, E-Commerce oder Betriebswirtschaftslehre.

Inhalt

- 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.

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

I.1.5 Seminar Informatik

M023 Seminar Informatik

| | |
|---------------------------------|---|
| Studiengang | Master-Studiengang Informatik |
| Kürzel | M023 |
| Bezeichnung | Seminar Informatik |
| Lehrveranstaltung(en) | M023a Seminar Informatik |
| Verantwortliche(r) | jeweiliger Dozent |
| Zuordnung zum Curriculum | Betriebswirtschaftslehre (Master) E-Commerce (Master) IT-Sicherheit (Master) Informatik (Master) |
| Verwendbarkeit | Die Fähigkeit, theoriegestützt zu arbeiten, wird in der Master-Thesis benötigt. |
| Semesterwochenstunden | 2 |
| ECTS | 5.0 |
| Voraussetzungen | Keine |
| Dauer | 1 |

Lernziele

Nach dem Seminar sind die Studierenden in der Lage, anspruchsvolle Themen eigenständig stärker theorieorientiert zu strukturieren und ihre Ausarbeitungen nach wissenschaftlichen Standards zu konzipieren. Im obligatorischen Vortrag können sie ihre Arbeitsergebnisse fundiert darlegen und im Diskurs kritisch diskutieren.

I.1.5.1 Seminar Informatik

| | |
|---------------------------------|--|
| Lehrveranstaltung | Seminar Informatik |
| Dozent(en) | jeweiliger Dozent |
| Hörtermin | 2 |
| Häufigkeit | jedes Semester |
| Lehrform | Seminar |
| Semesterwochenstunden | 2 |
| ECTS | 5.0 |
| Prüfungsform | Schriftl. Ausarbeitung (ggf. mit Präsentation) |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, Handout, interaktive Entwicklung und Diskussion von Modellen, Online-Aufbereitung, Software-demonstration, studentische Arbeit am Rechner, Tafel |

Lernziele

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.

Inhalt

Fachvorträge mit anschließender Gruppendiskussion.

Literatur

Recherche nach aufgabenbezogener Literatur, teilweise aufgabenspezifische Vorgabe einzelner Literaturquellen.

I.1.6 Workshop Cryptography

M009 Workshop Cryptography

| | |
|---------------------------------|---|
| Studiengang | Master-Studiengang Informatik |
| Kürzel | M009 |
| Bezeichnung | Workshop Cryptography |
| Lehrveranstaltung(en) | M009a Workshop Cryptography |
| Verantwortliche(r) | Prof. Dr. Gerd Beuster |
| Zuordnung zum Curriculum | IT Engineering (Master) IT-Sicherheit (Master) Informatik (Master) |
| Verwendbarkeit | Für dieses Modul sind Grundkenntnisse der diskreten Mathematik erforderlich. 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. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Die Studierenden benötigen das Wissen über diskrete Mathematik, das typischerweise in einem Bachelor-Studiengang in Informatik oder einem ähnlichen Bereich erworben wird. Die Studierenden müssen mit den gängigen Internet-Protokollen vertraut sein. Die Studierenden müssen über einige Grundkenntnisse in der Programmierung verfügen. |
| Dauer | 1 |

Lernziele

Im Kryptographie-Workshop erwerben die Studierenden Kenntnisse über die mathematischen Grundlagen der Kryptographie und deren praktische Anwendung. Nach Abschluss des Kurses sind die Studierenden in der Lage, kryptographische Verfahren im Kontext sicherer IT-Systeme einzusetzen und den Einsatz kryptographischer Verfahren in bestehenden Systemen zu evaluieren.

Dies umfasst sowohl software- als auch hardwarebasierte Kryptographie. Ein Schwerpunkt liegt auf der Kryptographie, die im Internet und für den E-Commerce eingesetzt wird. Die Studierenden wissen, wie die Vertraulichkeit und Integrität von persönlichen Daten und

Geschäftsdaten mit kryptographischen Mitteln sichergestellt werden kann. Basierend auf kryptographischen Systemen der realen Welt lernten die Studierenden, dass bei der Implementierung und Anwendung kryptographischer Methoden viele Nebenbedingungen berücksichtigt werden müssen.

I.1.6.1 Workshop Cryptography

| | |
|---------------------------------|---|
| Lehrveranstaltung | Workshop Cryptography |
| Dozent(en) | Gerd Beuster |
| Hörtermin | 2 |
| Häufigkeit | jährlich |
| Lehrform | Workshop |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Prüfungsform | Abnahme |
| Sprache | english |
| Lehr- und Medienform(en) | E-Learning, Softwaredemonstration, studentische Arbeit am Rechner |

Lernziele

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 Integritätserhaltung 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.

Inhalt

- Theorie der Kryptographie
 - Semantische Sicherheit
 - Unbrechbare Verschlüsselung und One Time Pad
 - Diffusion und Verwirrung
- 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
 - PRNG auf der Grundlage mathematischer Probleme
 - * Blum-Blum-Shub
- Hashing
 - Hashing-Algorithmen
 - * SHA 2
 - * Keccak
 - Nachrichten-Authentifizierung
 - * CMAC und HMAC
- Asymmetrische Kryptographie
 - Diffie-Hellman
 - RSA
 - Elliptische Kurven
 - Asymmetrische Verschlüsselung und digitale Signaturen
- Praktische Kryptographie: PGP und SSL
- Hardware-Kryptographie
 - Trusted Computing
 - Smartcards
 - Differenzleistungsanalyse

Literatur

- Stallings, William: Cryptography and Network Security : Principles and Practice. 7. Edition. London, UK: Pearson, 2017.
- Ferguson, Niels; Schneier, Bruce; Kohno, Tadayoshi: Cryptography Engineering : Design Principles and Practical Applications. Indianapolis (IN), USA: Wiley Publishing, 2010.

- Menezes, Alfred J.; van Oorschot, Paul C.; Vanstone, Scott A.: Handbook of Applied Cryptography. Boca Raton (FL), USA: CRC Press, 1996.
- Douglas R. Stinson: Cryptography : Theory and Practice. 3. Edition. Boca Raton (FL), USA: CRC Press, 2005.
- Lawrence C. Washington: Elliptic Curves : Number Theory and Cryptography. 2. Edition. Boca Raton (FL), USA: CRC Press, 2008.
- Joshua Davies: Implementing SSL/TLS Using Cryptography and PKI. Indianapolis (IN), USA: Wiley Publishing, 2011.
- Katz, Jonathan; Lindell, Yehuda: Introduction to Modern Cryptography. Boca Raton (FL), USA: CRC Press, 2007.
- Swenson, Christopher: Modern Cryptanalysis : Techniques for Advanced Code Breaking. Indianapolis (IN), USA: Wiley Publishing, 2008.
- Mao, Wenbo: Modern Cryptography: Theory and Practice, Upper Saddle River (NJ), USA: Prentice Hall, 2003.

I.1.7 Robotics

M018 Robotics

| | |
|---------------------------------|--|
| Studiengang | Master-Studiengang Informatik |
| Kürzel | M018 |
| Bezeichnung | Robotics |
| Lehrveranstaltung(en) | M018a Robotics |
| Verantwortliche(r) | Prof. Dr. Ulrich Hoffmann |
| Zuordnung zum Curriculum | Data Science & Artificial Intelligence (Master) IT Engineering (Master) Informatik (Master) |
| 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. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Voraussetzungen sind ein umfassendes Verständnis der Konzepte der Informationstechnologie und des Software-Engineering. Diese werden am besten durch ein vorhergehendes Studium der Informatik oder des IT-Engineering mit Schwerpunkt Medientechnik oder Computerarchitektur erreicht. Dieses Studium sollte mit einem Bachelor of Science in Informatik abgeschlossen worden sein. Es wird davon ausgegangen, dass die Studierenden in der Lage sein werden, selbständig in einem wissenschaftlichen Umfeld zu arbeiten. |
| Dauer | 1 |

Lernziele

Die Studierenden erwerben grundlegende Kompetenzen in ausgewählten Roboterkonzepten und Technologien.

Ein Schwerpunkt ist die Erforschung der Eigenschaften von mobilen und autonomen Systemen. Ausgehend von den Grundlagen der Robotik-Themen sammeln die Studierenden Erfahrungen mit Bewegungs- und Aktionsmodellierungskonzepten sowie mit intelligenten lernenden Sensoren als Grundlage für autonomes Roboterverhalten.

Eine beispielhafte Umsetzung eines der theoretisch vorgestellten Konzepte in einem schrit-tenorganisierten gruppenorientierten Projekt verbessert das Verständnis der vorliegenden Konzepte.

Die Studierenden erlangen insbesondere ein vertieftes Verständnis und können praktische Probleme kategorisieren und bewerten, die bei Roboteraktionen unter Anleitung von visueller Bildverarbeitung auftreten.

Darüber hinaus führt das Projekt zu einem verbesserten Präsentationsstil und einer verbesserten Präsentationstechnik sowie zu erweiterten Fähigkeiten zur freien Diskussion komplexer wissenschaftlicher Sachverhalte im Team.

I.1.7.1 Robotics

| | |
|---------------------------------|-----------------------------|
| Lehrveranstaltung | Robotics |
| Dozent(en) | Ulrich Hoffmann |
| Hörtermin | 2 |
| Häufigkeit | jährlich |
| Lehrform | mehrere Veranstaltungsarten |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Prüfungsform | Assessment |
| Sprache | english |
| Lehr- und Medienform(en) | |

Lernziele

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 Roboterhaltens.
- 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.

Inhalt

- Aufbau und Zusammensetzung von Robotern
 - Kinematik
 - Bewegung und Bewegter
 - 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

Literatur

- Blume, Dillmann: Frei Programmierbare Roboter, Vogel Verlag, 1981
- Stienecker: The KUKA Robot Programming Language, 2011

I.1.8 Praktikum Virtuelle Realität und Simulation

M062 Praktikum Virtuelle Realität und Simulation

| | |
|---------------------------------|---|
| Studiengang | Master-Studiengang Informatik |
| Kürzel | M062 |
| Bezeichnung | Praktikum Virtuelle Realität und Simulation |
| Lehrveranstaltung(en) | M062a Prakt. Virtuelle Realität und Simulation |
| Verantwortliche(r) | Prof. Dr. Christian-Arved Bohn |
| Zuordnung zum Curriculum | Informatik (Master) |
| 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. |
| Semesterwochenstunden | 5 |
| ECTS | 5.0 |
| Voraussetzungen | Tiefgehendes theoretisches Wissen der Inhalte der Module Computergrafik und Interaktive Systeme. Erste Erfahrungen in der Projektarbeit. |
| Dauer | 1 |

Lernziele

Durch praktische Anwendung der Techniken und Algorithmen aus der Vorlesung "Virtual und Augmented Reality" verfügen die Studierenden nach Abschluss des Moduls vertieftes Wissen in der Software-Entwicklung, tiefgehendes theoretisches Wissen der Inhalte der Module Computergrafik und Interaktive Systeme und weitere Erfahrungen in praktischer Projektarbeit.

I.1.8.1 Prakt. Virtuelle Realität und Simulation

| | |
|---------------------------------|--|
| Lehrveranstaltung | Prakt. Virtuelle Realität und Simulation |
| Dozent(en) | Christian-Arved Bohn |
| Hörtermin | 2 |
| Häufigkeit | jährlich |
| Lehrform | Übung/Praktikum/Planspiel |
| Semesterwochenstunden | 5 |
| ECTS | 5.0 |
| Prüfungsform | Abnahme |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, Softwaredemonstration, Tafel |

Lernziele

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 tiefergehendes Verständnis über typische VR-Installationen erlangt werden.

Inhalt

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.

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.

I.1.9 Moderne Software-Architekturen

M162 Moderne Software-Architekturen

| | |
|---------------------------------|--|
| Studiengang | Master-Studiengang Informatik |
| Kürzel | M162 |
| Bezeichnung | Moderne Software-Architekturen |
| Lehrveranstaltung(en) | M162a Moderne Software-Architekturen |
| Verantwortliche(r) | Prof. Dr. Ulrich Hoffmann |
| Zuordnung zum Curriculum | Informatik (Master) |
| Verwendbarkeit | Das Modul ist sinnvoll mit den Modulen "Distributed Systems" und "Learning and Softcomputing" kombinierbar. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Vorausgesetzt werden solide Kenntnisse in der Softwareentwicklung und ein Verständnis für den Einsatz großer Software-Systeme etwa im wirtschaftlichen Umfeld oder in der Forschung. |
| Dauer | 1 |

Lernziele

Studierende erwerben eingehende Kenntnisse im Bereich der Strukturierung großer Softwaresysteme, ihrer Entwicklung, ihrer Bereitstellung und ihres Betriebs. Durch konkreten Einsatz gängiger Softwarewerkzeuge kennen sie die praktischen Herausforderungen der Softwarebereitstellung und können umfangreiche und eigene Softwaresysteme selbständig bereitstellen und betreiben. Sie besitzen Kenntnisse und Fähigkeiten für die Messung an Softwaresystemen und können die Softwaresysteme anhand geeigneter Messergebnisse hin optimieren und skalieren.

Im praktischen Teil erwerben sie zusätzlich die Kompetenz arbeitsteilig in einer kleinen Arbeitsgruppe wissenschaftlich, selbständig an einer umfangreichen Aufgabe des Entwurfs und der Bereitstellung großer Systeme Kenntnisse zusammenzutragen und Lösungen zu erarbeiten sowie diese verständlich und strukturiert zu präsentieren.

I.1.9.1 Moderne Software-Architekturen

| | |
|---------------------------------|--|
| Lehrveranstaltung | Moderne Software-Architekturen |
| Dozent(en) | Ulrich Hoffmann |
| Hörtermin | 2 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Prüfungsform | Assessment |
| Sprache | deutsch |
| Lehr- und Medienform(en) | studentische Arbeit am Rechner, Tutorien |

Lernziele

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 Softwaresysteme 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 angemessenem Vortragsstil und geeigneter Präsentationstechniken nachvollziehbar darzustellen. In freier Diskussion können sie sich über komplexe wissenschaftliche Sachverhalte der großen Softwaresysteme und ihres Betriebs auseinandersetzen.

Inhalt

- 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

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

I.1.10 Berechenbarkeit und Verifikation

M029 Berechenbarkeit und Verifikation

| | |
|---------------------------------|--|
| Studiengang | Master-Studiengang Informatik |
| Kürzel | M029 |
| Bezeichnung | Berechenbarkeit und Verifikation |
| Lehrveranstaltung(en) | M029a Berechenbarkeit und Komplexität M029a Formale Spezifikation und Verifikation |
| Verantwortliche(r) | Prof. Dr. Sebastian Iwanowski |
| Zuordnung zum Curriculum | IT-Sicherheit (Master) Informatik (Master) |
| 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. |
| Semesterwochenstunden | 6 |
| ECTS | 5.0 |
| Voraussetzungen | Vorausgesetzt wird ein sehr gutes mathematisches Grundwissen, insbesondere der Logik und Mengenlehre. Die Teilnehmer sollten mit der Verwendung einer formalen Sprache vertraut sein und entsprechende Formeln verstehen. |
| Dauer | 1 |

Lernziele

Nach Abschluss des Moduls verfügen die Studierenden über einen theoretisch fundierten und umfassenden Überblick über die Möglichkeiten der Spezifikation von Lösung und Problemen. Sie kennen ferner die Grundlagen der klassischen Spezifikations- und Lösungsmethoden. Außerdem verfügen sie über eine theoretisch fundierte Beurteilungsfähigkeit bezüglich der Grenzen von Berechenbarkeit und effizienter Lösbarkeit.

I.1.10.1 Berechenbarkeit und Komplexität

| | |
|---------------------------------|---------------------------------|
| Lehrveranstaltung | Berechenbarkeit und Komplexität |
| Dozent(en) | Sebastian Iwanowski |
| Hörtermin | 1 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 3 |
| ECTS | 2.5 |
| Prüfungsform | Klausur / Mündliche Prüfung |
| Sprache | deutsch/englisch |
| Lehr- und Medienform(en) | Handout, Overheadfolien, Tafel |

Lernziele

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.

Inhalt

- 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

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

I.1.10.2 Formale Spezifikation und Verifikation

| | |
|---------------------------------|---|
| Lehrveranstaltung | Formale Spezifikation und Verifikation |
| Dozent(en) | Ulrich Hoffmann |
| Hörtermin | 1 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung mit integrierter Übung/Workshop/Assigm. |
| Semesterwochenstunden | 3 |
| ECTS | 2.5 |
| Prüfungsform | Klausur / Mündliche Prüfung |
| Sprache | deutsch/englisch |
| Lehr- und Medienform(en) | |

Lernziele

Die Studierenden ...

- erlangen fundierte Kenntnisse der mathematischen Grundlagen der formalen Spezifikation und Verifikation.
- beherrschen verschiedene Spezifikationsstile.
- bekommen einen Einblick in verschiedene formale Spezifikations Sprachen.
- 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.

Inhalt

- 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 Spezifikations Sprachen im Überblick

Literatur

- 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

I.1.11 Konzepte der Datenbanktechnologie

M027 Konzepte der Datenbanktechnologie

| | |
|---------------------------------|--|
| Studiengang | Master-Studiengang Informatik |
| Kürzel | M027 |
| Bezeichnung | Konzepte der Datenbanktechnologie |
| Lehrveranstaltung(en) | M027a Konzepte der Datenbanktechnologie M027b Übg. Konzepte der Datenbanktechnologie |
| Verantwortliche(r) | Dr. Michael Predeschly |
| Zuordnung zum Curriculum | Data Science & Artificial Intelligence (Master) E-Commerce (Master) IT-Sicherheit (Master) Informatik (Master) Wirtschaftsinformatik/IT-Management (Master) |
| Verwendbarkeit | Das Modul ist sinnvoll im Datenbanken-Curriculum zusammen mit den grundlegenden Modulen "Einführung in Datenbanken" und "Datenbanktheorie und -implementierung" aber auch den Programmier-einführungsmodulen ("Einführung in die Programmierung", "Programmstrukturen 1") zu kombinieren. Auch eine Kombination mit dem grundlegenden Modul "Systemmodellierung" ist ratsam. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Das Modul setzt solide Kenntnisse der Funktionsweise und des Aufbaus relationaler Datenbankmanagementsysteme voraus. Der praktische Anteil erfordert fortgeschrittene Fähigkeiten der objektorientierten Programmierung. |
| Dauer | 1 |

Lernziele

Nach Abschluss des Moduls besitzen die Studierenden fortgeschrittene Kenntnisse über Datenbanksysteme. Sie verfügen dabei über Wissen über relationaler Datenbanksysteme und über Datenbanksysteme, die auf alternativen Ansätzen (objekt-orientiert, objekt-relational, NoSQL, u., a.) basieren. Sie können deren Vor- und Nachteile abwägen. Die Studierenden sind in der Lage, sich kritisch mit den Möglichkeiten moderner Datenbanksysteme auseinanderzusetzen, diese geeignet einzuschätzen und praxisgerecht anzuwenden.

I.1.11.1 Konzepte der Datenbanktechnologie

| | |
|---------------------------------|---|
| Lehrveranstaltung | Konzepte der Datenbanktechnologie |
| Dozent(en) | Michael Predeschly |
| Hörtermin | 1 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 2 |
| ECTS | 3.0 |
| Prüfungsform | Klausur / Mündliche Prüfung |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, E-Learning, Gastreferenten, Online-Aufbereitung, Softwaredemonstration, Tafel, Tutorien |

Lernziele

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

Inhalt

- 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

Literatur

- 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

I.1.11.2 Übg. Konzepte der Datenbanktechnologie

| | |
|---------------------------------|--|
| Lehrveranstaltung | Übg. Konzepte der Datenbanktechnologie |
| Dozent(en) | Michael Predeschly |
| Hörtermin | 1 |
| Häufigkeit | jährlich |
| Lehrform | Übung/Praktikum/Planspiel |
| Semesterwochenstunden | 2 |
| ECTS | 2.0 |
| Prüfungsform | Abnahme |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, E-Learning, Gastreferenten, Online-Aufbereitung, Tafel |

Lernziele

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

Inhalt

Vorlesungsbegleitende praktische Übungen zu Objektrelationalem Mapping und anderen alternativen Persistenzansätzen.

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

Literatur

- siehe Vorlesung
- diverse Online-Quellen

I.1.12 Distributed Systems

M035 Distributed Systems

| | |
|---------------------------------|---|
| Studiengang | Master-Studiengang Informatik |
| Kürzel | M035 |
| Bezeichnung | Distributed Systems |
| Lehrveranstaltung(en) | M035a Distributed Systems M035b Tutorial: Distributed Systems |
| Verantwortliche(r) | Prof. Dr. Ulrich Hoffmann |
| Zuordnung zum Curriculum | IT Engineering (Master) IT-Sicherheit (Master) Informatik (Master) Wirtschaftsinformatik/IT-Management (Master) |
| Verwendbarkeit | Das Modul kann gut mit den Modulen "Funktionales Programmieren" und "Aktuelle Entwicklungen in der Informatik" sowie mit dem "Seminar-Master" kombiniert werden.. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Die praktischen Übungen setzen fortgeschrittene Programmierfähigkeiten voraus. Darüber hinaus setzt das Modul solide Kenntnisse der Internetarchitektur und -struktur sowie Grundkenntnisse der Ablauforganisation in Unternehmen voraus. |
| Dauer | 1 |

Lernziele

Die Studierenden erwerben vertiefte Kenntnisse über technische Aspekte verteilter Systeme sowie deren Anwendungsgebiete in kommerziellen Kontexten. Sie erleben und diskutieren die technologischen inhärenten Probleme verteilter Systeme und sind dadurch in der Lage, sich den Herausforderungen verteilter Systeme, wie etwa Fragen der IT-Sicherheit oder der verschlüsselten Kommunikation, zu stellen und mit ihnen umzugehen. Sie kennen die Architektur und die wichtigsten Algorithmen in verteilten Systemen sowie die Prozesse in Entwicklung und Administration, die zu erfolgreichen verteilten Produkten führen. Sie sind in der Lage, verteilte Systeme in verschiedenen Programmierparadigmen zu programmieren.

I.1.12.1 Distributed Systems

| | |
|---------------------------------|-----------------------------|
| Lehrveranstaltung | Distributed Systems |
| Dozent(en) | Ulrich Hoffmann |
| Hörtermin | 1 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 2 |
| ECTS | 3.0 |
| Prüfungsform | Klausur / Mündliche Prüfung |
| Sprache | english |
| Lehr- und Medienform(en) | |

Lernziele

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 Dienstleistungs-vermittlung 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.

Inhalt

- 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

Literatur

- 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

I.1.12.2 Tutorial: Distributed Systems

| | |
|---------------------------------|-------------------------------|
| Lehrveranstaltung | Tutorial: Distributed Systems |
| Dozent(en) | Ulrich Hoffmann |
| Hörtermin | 1 |
| Häufigkeit | jährlich |
| Lehrform | Übung/Praktikum/Planspiel |
| Semesterwochenstunden | 2 |
| ECTS | 2.0 |
| Prüfungsform | Abnahme |
| Sprache | english |
| Lehr- und Medienform(en) | |

Lernziele

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.

Inhalt

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

Literatur

- siehe Vorlesung
- Zahlreiche Online-Ressourcen

I.1.13 Methoden der Künstlichen Intelligenz

M033 Methoden der Künstlichen Intelligenz

| | |
|---------------------------------|--|
| Studiengang | Master-Studiengang Informatik |
| Kürzel | M033 |
| Bezeichnung | Methoden der Künstlichen Intelligenz |
| Lehrveranstaltung(en) | M033a Methoden der Künstlichen Intelligenz |
| Verantwortliche(r) | Prof. Dr. Gerd Beuster |
| Zuordnung zum Curriculum | Data Science & Artificial Intelligence (Master) IT-Sicherheit (Master) Informatik (Master) Wirtschaftsinformatik/IT-Management (Master) |
| Verwendbarkeit | Das Modul setzt voraus, dass die Studierenden die grundlegenden Algorithmen der Informatik und Grundlagen diskreter algebraischer Strukturen kennen. Die im Modul erworbenen Fähigkeiten können überall dort verwendet werden, wo autonom handelnde Agenten benötigt werden. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Die Studierenden benötigen die in einem Bachelor-Studium der Informatik oder einem ähnlichen Studium erworbenen Kenntnisse über diskrete algebraische Strukturen und grundlegende Algorithmen der Informatik. Die Studierenden verfügen über Programmierkenntnisse. |
| Dauer | 1 |

Lernziele

Nach Abschluss des Moduls verfügen die Studierenden über das Wissen über grundsätzliche Verfahrensweisen der Künstlichen Intelligenz im weiteren Sinne. Sie verfügen über einen umfassenden Überblick der theoretischen Grundlagen sowie über ein gutes Verständnis für die Implementierung ausgewählter Verfahren. Der Schwerpunkt liegt hierbei in der symbolischen Künstlichen Intelligenz und Methoden der formalen Logik. 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.

I.1.13.1 Methoden der Künstlichen Intelligenz

| | |
|---------------------------------|--|
| Lehrveranstaltung | Methoden der Künstlichen Intelligenz |
| Dozent(en) | Gerd Beuster |
| Hörtermin | 1 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung mit integrierter Übung/Workshop/Assignm. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Prüfungsform | Klausur / Mündliche Prüfung |
| Sprache | deutsch/englisch |
| Lehr- und Medienform(en) | E-Learning, interaktive Entwicklung und Diskussion von Modellen, Softwaredemonstration, studentische Arbeit am Rechner |

Lernziele

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.

Inhalt

- Einführung in die Künstliche Intelligenz
- Intelligente Agenten
- Suchverfahren
- Aussagenlogik
- Logikbasierte autonome Agenten
- Prädikatenlogik
- Grenzen der Prädikatenlogik
- Logikprogrammierung
- Prädikatenlogisches Planen

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. Cambridge: Cambridge University Press, 2010.
- Norvig, Peter; Russell, Stuart: Artificial Intelligence : A Modern Approach. 3. Auflage. Upper Saddle River (NJ), USA: Prentice Hall, 2009.
- 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.

I.1.14 Projekt Informatik

M048 Projekt Informatik

| | |
|---------------------------------|---|
| Studiengang | Master-Studiengang Informatik |
| Kürzel | M048 |
| Bezeichnung | Projekt Informatik |
| Lehrveranstaltung(en) | M048a Projekt Informatik |
| Verantwortliche(r) | jeweiliger Dozent |
| Zuordnung zum Curriculum | Informatik (Master) |
| Verwendbarkeit | Das Modul dient als Vorbereitung auf die Master-Thesis. |
| Semesterwochenstunden | 0 |
| ECTS | 5.0 |
| Voraussetzungen | Gute Vorkenntnisse in dem Fachgebiet, in dem das Projekt abläuft. |
| Dauer | 1 |

Lernziele

Innerhalb eines nichttrivialen Informatik-Projekts mit anspruchsvoller Zielstellung üben die Studierenden das Arbeiten in einem vorgegebenen technischen Umfeld, das in die Arbeitsgruppen und Labore der Hochschule oder in konkreten Unternehmenskontext integriert ist. Durch eine möglichst praxisnahe Ausrichtung der Arbeitsweisen sammeln die Studierenden Erfahrungen im Projekt-Management und den Bereichen Projektplanung, Koordination, Aufgabenaufteilung, Zeitmanagement, Delegation und Controlling und machen sich mit den dort auftretenden Herausforderungen vertraut.

I.1.14.1 Projekt Informatik

| | |
|---------------------------------|--|
| Lehrveranstaltung | Projekt Informatik |
| Dozent(en) | jeweiliger Dozent |
| Hörtermin | 1 |
| Häufigkeit | jedes Semester |
| Lehrform | Projekt |
| Semesterwochenstunden | 0 |
| ECTS | 5.0 |
| Prüfungsform | Schriftl. Ausarbeitung (ggf. mit Präsentation) |
| Sprache | deutsch |
| Lehr- und Medienform(en) | |

Lernziele

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.

Inhalt

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.

Literatur

Themenabhängig

I.1.15 Digitale Kommunikationssysteme und Reconfigurable Computing

M042 Digitale Kommunikationssysteme und Reconfigurable Computing

| | |
|---------------------------------|---|
| Studiengang | Master-Studiengang Informatik |
| Kürzel | M042 |
| Bezeichnung | Digitale Kommunikationssysteme und Reconfigurable Computing |
| Lehrveranstaltung(en) | M042a Digitale Kommunikationssysteme M042b Reconfigurable Computing M042b Prakt. Reconfigurable Computing |
| Verantwortliche(r) | Prof. Dr. Sergei Sawitzki |
| Zuordnung zum Curriculum | Informatik (Master) |
| 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. |
| Semesterwochenstunden | 6 |
| ECTS | 5.0 |
| Voraussetzungen | Grundkenntnisse der Digitaltechnik und Rechnerstrukturen sowie Entwurfsmethoden digitaler Systeme, Grundkompetenzen aus der Systemtheorie sowie Funktionaltransformationen. |
| Dauer | 1 |

Lernziele

Das Modul vermittelt die wesentlichen Konzepte, Methoden und Anwendungen von Reconfigurable Computing als eines der Entwurfs- und Implementierungsparadigmen im modernen Systementwurf. Der Vorlesungsteil dient der Einführung in die Begriffswelt, die Denkweisen

und die Konzepte des rekonfigurierbaren Entwurfs, während im anschließenden Praktikum ein zusätzlicher Erkenntnisgewinn durch Implementierung und Optimierung einer komplexen Anwendung erreicht wird. Während die Anwendungsabbildung auf rekonfigurierbare Hardwarestrukturen mit der Markteinführung erster FPGA-Schaltungen im Jahre 1985 erstmals technologischen Aufwind bekam, so hat insbesondere die Entwicklung in den letzten 15-20 Jahren dieses Fachgebiet als Forschungs- und Entwicklungsumfeld fest etabliert. Somit lernen die Studierenden die modernsten Entwurfsmethodiken (bis hin zum Reconfigurable-System-on-Chip) kennen und gewinnen einige Anreize für eventuelle spätere berufliche Entwicklung. Im Teil "Digitale Kommunikationssysteme" werden Kompetenzen und Methoden vermittelt, die unabdingbar sind, um die Funktionsweise, Qualitätskriterien, Parameter und Standards digitaler Kommunikation einordnen und sinnvoll einsetzen zu können. Die theoretischen Grundlagen werden mit vielen Beispielen real existierenden Systeme untermauert.

I.1.15.1 Digitale Kommunikationssysteme

| | |
|---------------------------------|---|
| Lehrveranstaltung | Digitale Kommunikationssysteme |
| Dozent(en) | Sergei Sawitzki |
| Hörtermin | 1 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 2 |
| ECTS | 1.0 |
| Prüfungsform | Mündliche Prüfung |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, Handout, Softwaredemonstration, Tafel |

Lernziele

Die Studierenden ...

- erlangen eine vertiefte Kenntnis moderner Übertragungssysteme, insbesondere des Aufbaus und der Funktionsweise von Basisband Transceivern
- lernen verschiedene Implementierungsaspekte von digitalen Kommunikationssystemen kennen
- erlangen das Verständnis der Abhängigkeiten zwischen verschiedenen Systemparametern und der erreichbaren Übertragungsqualität
- lernen Systeme und Übertragungskanäle zu bewerten und zu modellieren
- erlangen das Verständnis von Qualitätskriterien digitaler Kommunikationssysteme und Einflussfaktoren bei digitaler Datenübertragung
- erlangen die Fähigkeit, digitale Übertragungsstandards zu interpretieren und auf der Ebene der Systemarchitektur (bis hin zur algorithmischen Ebene) zu spezifizieren und zu entwerfen

Inhalt

- 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)

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

I.1.15.2 Reconfigurable Computing

| | |
|---------------------------------|---|
| Lehrveranstaltung | Reconfigurable Computing |
| Dozent(en) | Sergei Sawitzki |
| Hörtermin | 1 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 2 |
| ECTS | 1.0 |
| Prüfungsform | Mündliche Prüfung |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, Handout, Softwaredemonstration, Tafel |

Lernziele

Die Studierenden ...

- lernen rekonfigurierbare Rechnersysteme als Entwurfsvariante des modernen Systementwurfs kennen
- vertiefen die Kenntnisse über die schaltungstechnische Basis und Hardware-Plattformen des Reconfigurable Computing
- erlangen die Fähigkeit, Vor- und Nachteile einer rekonfigurierbaren Implementierung eines Systems realistisch abschätzen zu können

Inhalt

- 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

Literatur

- 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

I.1.15.3 Prakt. Reconfigurable Computing

| | |
|---------------------------------|---------------------------------|
| Lehrveranstaltung | Prakt. Reconfigurable Computing |
| Dozent(en) | Sergei Sawitzki |
| Hörtermin | 1 |
| Häufigkeit | jährlich |
| Lehrform | Praktikum |
| Semesterwochenstunden | 2 |
| ECTS | 3.0 |
| Prüfungsform | Mündliche Prüfung |
| Sprache | deutsch |
| Lehr- und Medienform(en) | studentische Arbeit am Rechner |

Lernziele

Die Studierenden vertiefen die in der Lehrveranstaltung "Reconfigurable Computing" erworbenen Kenntnisse durch Abbildung einer technischen Aufgabenstellung auf ein konventionelles und ein rekonfigurierbares Rechnersystem und quantitative Analyse der Ergebnisse.

Inhalt

- Einführung
 - Vorstellung der Aufgabenstellung
 - Einarbeitung in die Entwurfswerkzeuge
- Umsetzung und Dokumentation der Aufgabe

Literatur

- 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.)

I.1.16 Fotorealismus und Simulation

M044 Fotorealismus und Simulation

| | |
|---------------------------------|--|
| Studiengang | Master-Studiengang Informatik |
| Kürzel | M044 |
| Bezeichnung | Fotorealismus und Simulation |
| Lehrveranstaltung(en) | M044a Fotorealismus und Simulation M044a Visualisierung |
| Verantwortliche(r) | Prof. Dr. Christian-Arved Bohn |
| Zuordnung zum Curriculum | Informatik (Master) |
| 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. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Voraussetzung ist das Verständnis der grundlegenden Algorithmen der Computergrafik. |
| Dauer | 1 |

Lernziele

Studierende erlernen fortgeschrittenen Algorithmen der Computergrafik, im Einzelnen sind dies die Grundlagen der physikalischen Simulation und fortgeschrittene Themen der geometrischen Modellierung.

I.1.16.1 Fotorealismus und Simulation

| | |
|---------------------------------|---|
| Lehrveranstaltung | Fotorealismus und Simulation |
| Dozent(en) | Christian-Arved Bohn |
| Hörtermin | 1 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 2 |
| ECTS | 2.0 |
| Prüfungsform | Mündliche Prüfung |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, Handout, Softwaredemonstration, Tafel |

Lernziele

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.

Inhalt

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.

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.

I.1.16.2 Visualisierung

| | |
|---------------------------------|---|
| Lehrveranstaltung | Visualisierung |
| Dozent(en) | Christian-Arved Bohn |
| Hörtermin | 1 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 2 |
| ECTS | 3.0 |
| Prüfungsform | Mündliche Prüfung |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, Handout, Softwaredemonstration, Tafel |

Lernziele

Studierende erlangen das Verständnis

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

Inhalt

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, Lindenmayersysteme, 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.

Literatur

- 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.

I.1.17 Aktuelle Entwicklungen in der Informatik

M010 Aktuelle Entwicklungen in der Informatik

| | |
|---------------------------------|---|
| Studiengang | Master-Studiengang Informatik |
| Kürzel | M010 |
| Bezeichnung | Aktuelle Entwicklungen in der Informatik |
| Lehrveranstaltung(en) | M010a Workshop Aktuelle Entwicklungen in der Informatik |
| Verantwortliche(r) | Prof. Dr. Ulrich Hoffmann |
| Zuordnung zum Curriculum | Informatik (Master) |
| Verwendbarkeit | Das Modul ist mit den Modulen "Distributed Systems" und "Funktionale Programmierung" aber auch mit dem Modul "Seminar Master" gut kombinierbar. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Dieses Modul setzt solide Informatik Grundkenntnissen und Fähigkeiten voraus, wie sie durch ein erfolgreiches Bachelor-Studium einer Informatik-Disziplin erworben werden können. |
| Dauer | 1 |

Lernziele

In diesem Modul erwerben die Studierenden umfangreiche Kenntnisse und Fähigkeit in ausgewählten, fortgeschrittenen Themenbereichen der Informatik. Als Themen werden dabei aktuelle Entwicklungen in der angewandten Informatik eingehend betrachtet. Durch den praktischen Umgang mit den jeweils einschlägigen Softwaresystemen gewinnen die Studierenden zudem auch praktische Kompetenz, verschiedene Lösungsverfahren einzuordnen, zu beurteilen und sie zur Problemlösung im jeweiligen Themenbereich anwenden zu können.

I.1.17.1 Workshop Aktuelle Entwicklungen in der Informatik

| | |
|---------------------------------|---|
| Lehrveranstaltung | Workshop Aktuelle Entwicklungen in der Informatik |
| Dozent(en) | Frank Huch |
| Hörtermin | 1 |
| Häufigkeit | jährlich |
| Lehrform | Workshop |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Prüfungsform | Abnahme |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Softwaredemonstration, studentische Arbeit am Rechner |

Lernziele

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.

Inhalt

themenabhängig

Literatur

themenabhängig

I.1.18 Master-Thesis

M050 Master-Thesis

| | |
|---------------------------------|--|
| Studiengang | Master-Studiengang Informatik |
| Kürzel | M050 |
| Bezeichnung | Master-Thesis |
| Lehrveranstaltung(en) | M050a Master-Thesis |
| Verantwortliche(r) | jeweiliger Dozent |
| Zuordnung zum Curriculum | Betriebswirtschaftslehre (Master) Data Science & Artificial Intelligence (Master) E-Commerce (Master) IT-Sicherheit (Master) Informatik (Master) Sustainable & Digital Business Management (Master) Wirtschaftsinformatik/IT-Management (Master) Wirtschaftsingenieurwesen (Master) |
| Verwendbarkeit | Keine |
| Semesterwochenstunden | 0 |
| ECTS | 28.0 |
| Voraussetzungen | Voraussetzung für die Master-Thesis ist der Stoff aus den vorangegangenen beiden Semestern, insbesondere der Veranstaltungen, die einen Bezug zur Themenstellung der Arbeit haben. |
| Dauer | 1 |

Lernziele

In der Masterthesis zeigen die Studierenden, dass sie in der Lage sind, komplexe Aufgabenstellungen mit wissenschaftlich methodischer Vorgehensweise selbstständig und zielorientiert zu erarbeiten. Sie sind befähigt, Problemstellungen im größeren Kontext zu verorten, die fachlichen Zusammenhänge zu vernetzen und die gewonnenen Erkenntnisse argumentativ überzeugend darzustellen und zu präsentieren.

I.1.18.1 Master-Thesis

| | |
|---------------------------------|--|
| Lehrveranstaltung | Master-Thesis |
| Dozent(en) | jeweiliger Dozent |
| Hörtermin | 3 |
| Häufigkeit | jedes Semester |
| Lehrform | Thesis |
| Semesterwochenstunden | 0 |
| ECTS | 28.0 |
| Prüfungsform | Schriftl. Ausarbeitung (ggf. mit Präsentation) |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, Tafel |

Lernziele

Die Studierenden sind in der Lage ...

- komplexe Aufgabenstellungen selbständig zu erarbeiten.
- Problemstellungen im größeren Kontext zu verorten.
- wissenschaftliche Methoden für die Problemlösung einzusetzen.
- Ergebnisse überzeugend darzustellen.

Inhalt

themenabhängig

Literatur

themenabhängig

I.1.19 Master-Kolloquium

M058 Master-Kolloquium

| | |
|---------------------------------|--|
| Studiengang | Master-Studiengang Informatik |
| Kürzel | M058 |
| Bezeichnung | Master-Kolloquium |
| Lehrveranstaltung(en) | M058a Kolloquium |
| Verantwortliche(r) | jeweiliger Dozent |
| Zuordnung zum Curriculum | Betriebswirtschaftslehre (Master) Data Science & Artificial Intelligence (Master) E-Commerce (Master) IT-Sicherheit (Master) Informatik (Master) Sustainable & Digital Business Management (Master) Wirtschaftsinformatik/IT-Management (Master) Wirtschaftsingenieurwesen (Master) |
| Verwendbarkeit | Keine |
| Semesterwochenstunden | 0 |
| ECTS | 2.0 |
| Voraussetzungen | Zulassungsvoraussetzung zum Kolloquium ist eine mit mindestens "ausreichend" bewertete Master-Thesis. |
| Dauer | 1 |

Lernziele

Die Studierenden präsentieren ihre Arbeitsergebnisse überzeugend vor dem Prüfungsausschuss. Sie beherrschen das Instrument der freien Rede, argumentieren schlüssig und beweisführend. In einer anschließenden fächerübergreifenden mündlichen Prüfung verteidigen sie ihre Arbeitsergebnisse und erweisen sich in der Diskussion als problemvertraut.

I.1.19.1 Kolloquium

| | |
|---------------------------------|-----------------------|
| Lehrveranstaltung | Kolloquium |
| Dozent(en) | verschiedene Dozenten |
| Hörtermin | 3 |
| Häufigkeit | jedes Semester |
| Lehrform | Kolloquium |
| Semesterwochenstunden | 0 |
| ECTS | 2.0 |
| Prüfungsform | Kolloquium |
| Sprache | deutsch |
| Lehr- und Medienform(en) | |

Lernziele

Die Studierenden ...

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

Inhalt

- 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

Literatur

themenabhängig

| | |
|------------------------|--|
| Dokumenttyp | Modulhandbuch |
| Abschlusstyp | Master |
| Studiengangname | Informatik |
| Ordnungsnummer | 20.0 |
| Setzdatum | 16. Dezember 2021 |
| git | ja |
| git-commit | 99b702a1 (lokale Änderungen vorhanden) |