



Ostfalia
Hochschule für angewandte
Wissenschaften

Fakultät Informatik

Modulhandbuch

M.Sc. Informatik (PO25)

Version 1 vom 04.06.2024

Salzgitter · Suderburg · **Wolfenbüttel** · Wolfsburg

Inhaltsverzeichnis

1	Einleitung	3
1.1	Inhalte und Studienverlauf	3
1.2	Hinweise zum Aufbau der Modulbeschreibungen	4
1.3	Weitere Informationen	6
2	Modulbeschreibungen	7
2.1	Grundlagenteil / Kompetenzteil Software Engineering	7
2.1.1	Architekturen moderner Informationssysteme (M)	7
2.1.2	Cloud Native Computing (M)	9
2.1.3	Effiziente Algorithmen (M)	10
2.1.4	Entwicklung großer Anwendungssysteme (M)	11
2.1.5	Formale Methoden (M)	12
2.1.6	IT-Management (M)	13
2.1.7	Modellgetriebene Softwareentwicklung (M)	14
2.1.8	User Interfaces für Mobile Systeme (M)	15
2.1.9	Werkzeuge der Java-Plattform (M)	16
2.2	Grundlagenteil / Kompetenzteil Data Science	17
2.2.1	Statistische Methoden (M)	17
2.2.2	Heuristische Suche (M)	18
2.2.3	Neuronale Netze und Deep Learning (M)	19
2.2.4	Automatische Sprachverarbeitung (M)	20
2.2.5	Datenbanktechnologien (M)	21
2.2.6	Big Data (M)	22
2.2.7	Angewandte Kryptographie (M)	23
2.2.8	Machine Learning (M)	24
2.2.9	Stream Processing (M)	25
2.3	Grundlagenteil / Kompetenzteil Systems & Computer Engineering	27
2.3.1	Automotive Systems (M)	27
2.3.2	Autonomous Systems (M)	28
2.3.3	Robotik (M)	29
2.3.4	Robuste Systeme (M)	30
2.3.5	Simulation und Verifikation (M)	32
2.3.6	Smart IoT (M)	33
2.3.7	Softwareintensive Systeme in der Mobilität (M)	34
2.3.8	Human-Robot Interaction (M)	35
2.4	Grundlagenteil überfachliche Kompetenzen	36
2.4.1	Entwicklung digitaler Geschäftsmodelle (M)	36
2.4.2	Gesprächs- und Verhandlungsführung - Leitung von Arbeitsgruppen (M)	37
2.5	Grundlagenteil ohne Schwerpunktzuordnung	38
2.5.1	Programmierparadigmen C++ (M)	38
2.6	Qualifikationsmodule	39
2.6.1	Masterseminar (M)	39
2.6.2	Masterprojekt (M)	40
2.6.3	Großes Masterprojekt (M)	41
2.7	Masterarbeit mit Kolloquium	42
2.7.1	Masterarbeit mit Kolloquium (M)	42
3	Modulübersichtstabelle	43
4	Änderungshistorie	46

1 Einleitung

1.1 Inhalte und Studienverlauf

Dieses Modulhandbuch beschreibt alle Lehrveranstaltungen des Studienganges **Informatik (M. Sc.)**. Für jede Veranstaltung werden die Lernziele, eine grobe Inhaltsübersicht sowie weitere Informationen wie Voraussetzungen, Prüfungsform, Arbeitsaufwand und Leistungspunkte angegeben.

Der Studienverlaufsplan ist in Abbildung 1.1 dargestellt:

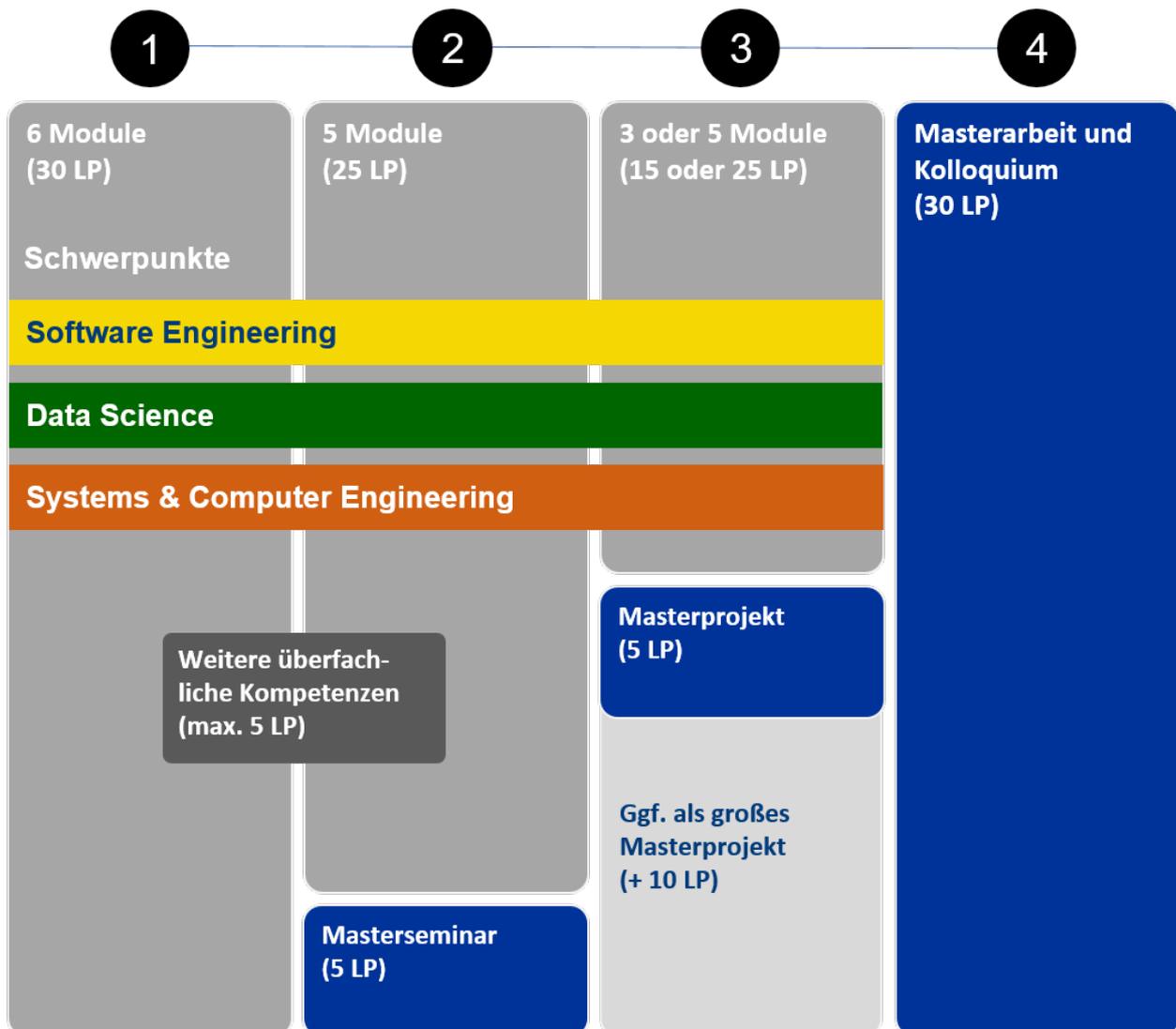


Abbildung 1.1: Studienverlauf ohne Berücksichtigung des *Großen Masterprojektes* (15 LP) als Alternative zum *Masterprojekt* (5 LP)

Der Studiengang besteht im Wesentlichen aus Wahlpflichtmodulen aus einem **Grundlagenteil/Kompetenzteil** mit und ohne Schwerpunktzuordnung sowie überfachlichen Kompetenzen (siehe Abbildung 1.1, grau hinterlegt). Hinzu kommen die **Qualifikationsmodule** Masterseminar und Masterprojekt sowie die Masterarbeit mit Kolloquium (in der Abbildung petrolgrün hinterlegt).

Studierende haben dabei die folgenden Optionen:

- Neben den Informatik-Wahlpflichtmodulen können maximal zwei Wahlpflichtmodule aus dem **Grundlagenteil überfachliche Kompetenzen** gewählt werden. Auf Antrag können hier auch Fächer aus

dem Angebot anderer Masterstudiengänge oder aus dem Masterangebot des Career Service eingebracht werden.

- Die Module aus dem **Grundlagenteil / Kompetenzteil** sind Schwerpunkten zugeordnet. Damit können die Studierenden ihrem Masterstudium ein spezifisches Kompetenzprofil geben. Auf Antrag können bis zu zwei Schwerpunkte auf dem Abschlusszeugnis eingetragen werden. Dazu müssen mindestens sieben Module aus dem Schwerpunkt bestanden sein.
- In Abstimmung mit Erst- und Zweitbetreuer/-in können stark forschungsinteressierte Studierende statt des *Masterprojektes* und zweier regulärer Wahlpflichtmodule der Informatik auch ein *Großes Masterprojekt* im Umfang von 15 Leistungspunkten belegen.

Es sind die folgenden Studienschwerpunkte vorgesehen:

- Software Engineering (SE)
- Data Science (DS)
- Systems & Computer Engineering (SCE)

1.2 Hinweise zum Aufbau der Modulbeschreibungen

Grundsätzlich können Module in **Lehrveranstaltungen** unterteilt werden, wobei die meisten Module aus einer einzigen gleichnamigen Lehrveranstaltung bestehen.

Die einzelnen **Prüfungsformen** werden in der Prüfungsordnung beschrieben. Abweichungen von der in diesem Handbuch angegebenen *Prüfungsform* werden gegebenenfalls zu Anfang des Semesters in der Lehrveranstaltung bekanntgegeben. Es werden verschiedene Prüfungsformen verwendet (siehe Tabelle 1.1). Im Falle einer Portfolioprüfung wird für die einzelnen Module explizit angegeben, ob auch eine Klausur Bestandteil der Prüfungsform ist. Für die Seitenangaben in den genannten Prüfungsumfängen wird die Formatvorlage der Ostfalia für Abschlussarbeiten zugrunde gelegt.

Prüfungsform	Prüfungsumfang
Klausur (K)	Prüfungsdauer: 1 - 3 Stunden
Mündliche Prüfung (M)	Prüfungsdauer: 15 - 45 Minuten
Rechnergestützte Prüfung (RP)	Prüfungsdauer: 1 - 3 Stunden
Praxisbericht (PB)	Umfang: 15 - 30 Seiten
Experimentelle Arbeit / Projektarbeit (EA,PA)	Bearbeitungszeit: 10 - 80 Stunden
Hausarbeit (H)	Umfang: 5 - 30 Seiten
Portfolioprüfung (PF)	Bearbeitungszeit: 10 - 80 Stunden
Erstellung und Dokumentation von Rechnerprogrammen (ED)	Bearbeitungszeit: 10 - 80 Stunden, Dokumentation: 1 - 10 Seiten

Tabelle 1.1: Prüfungsformen

In der Beschreibung der Module wird neben der Prüfungsform für Klausuren auch die Dauer der Klausur in Stunden angegeben, also z.B. *K1,5h*. Für zusammengesetzte Prüfungsformen werden die einzelnen Anteile mit einem + verknüpft, wobei die prozentualen Anteile in Klammern angegeben werden, also z.B. *EA(30%)+K1,5h(70%)*. Darüber hinaus können auch alternative Prüfungsformen angegeben werden. Diese werden durch ein / getrennt. Für eine bessere Lesbarkeit werden die Alternativen zudem in Klammern angegeben, also z.B. *(EA(30%)+K1,5h(70%))/(M)*, womit entweder eine mündliche Prüfung *oder* eine Kombination aus experimenteller Arbeit und Klausur vorgesehen werden kann.

Die Angabe der **Leistungspunkte** folgt dem *European Credit Transfer and Accumulation System* (ECTS). Das ECTS ermöglicht Studierenden die einfache Anerkennung von im In- und Ausland erbrachten Studienleistungen. Dabei werden jedem Modul eine bestimmte Anzahl an Leistungspunkten zugeordnet, die bei erfolgreichem Abschluss einer Veranstaltung angerechnet werden.

Die studentische Arbeitsbelastung liegt pro ECTS-Punkt bei einem durchschnittlichen **Aufwand** von 30 h und setzt sich aus der Kontaktzeit mit den Dozenten/-innen, der Zeit für das Selbststudium und dem Zeitaufwand für die Prüfungsvorbereitung zusammen. Die Arbeit an Aufgaben der Prüfungsformen Referat (R), Experimentelle Arbeit/Projektarbeit (EA, PA), Hausarbeit (H), Portfolioprüfung (PF) und Erstellung und Dokumentation (ED) erfolgt typischerweise auch im Rahmen des Selbststudiums. Die Lehrenden stellen dabei sicher, dass die studentische Arbeitsbelastung den für das Modul vorgesehenen Gesamtrahmen von 30 h/ECTS-Punkt nicht überschreitet.

Im Feld **Verwendbarkeit** sind die Studiengänge und PO-Versionen angegeben, in denen das jeweilige Modul eingebracht werden kann. Bei Angabe der Bachelorstudiengänge *Informatik* bzw. *Wirtschaftsinformatik* (unabhängig von der PO-Version) sind grundsätzlich auch die jeweiligen Pendanten *Informatik im Praxisverbund* bzw. *Wirtschaftsinformatik im Praxisverbund* inkludiert.

Im Feld **empfohlene Voraussetzungen** wird nochmals explizit angegeben, welche Vorkenntnisse (zum Beispiel aus anderen Lehrveranstaltungen) für einen erfolgreichen Besuch der Veranstaltung notwendig sind. Grundsätzlich werden jedoch auch ohne eine solche Angabe Vorkenntnisse aus den im Curriculum vorgelegten Modulen vorausgesetzt. Die hier angegebenen Voraussetzungen haben Empfehlungscharakter und stellen keine formale Anforderung für den Besuch des Moduls oder das Ablegen der Prüfung dar. Solche formalen Anforderungen sind allerdings als Teil der Prüfungsordnung möglich.

Aus der Angabe der **Lehr- und Lernformen** ist ersichtlich, wie viele Semesterwochenstunden (SWS) im jeweiligen Format gelehrt werden. Dabei wird nur die Kontaktzeit berücksichtigt. Eine SWS entspricht 45 Minuten. Grundsätzlich werden die in Tabelle 1.2 beschriebenen Lehr- und Lernformen verwendet.

Lehrform	Beschreibung
Vorlesung (V)	Vorlesung (im Wesentlichen Lehrvortrag) in Interaktion mit den Studierenden
Übung (Ü)	Bearbeitung von praxisnahen Übungsaufgaben in Einzel- und Teamarbeit
Labor (L)	Bearbeitung von Labor- und Praktikumsaufgaben am Rechner bzw. an Hardware in Einzel- oder Teamarbeit
Seminaristischer Unterricht (SU)	Vorlesung (Lehrvortrag und -gespräch) in enger Interaktion mit den Studierenden, die eine aktive Beteiligung aller Studierenden erfordert und neben reiner Wissensvermittlung auch Übungs- und Laboranteile integriert
Flipped Classroom (FC)	Vorbereitung der Lehrinhalte durch die Studierenden anhand von Materialien wie z. B. Lehrvideos oder -texte; Nutzung der Präsenzzeiten für Fragen, Vertiefung, Übungs- und Laboranteile
Projekt (P)	Bearbeitung einer umfangreichen (mehrwöchigen) praxisnahen Aufgabenstellung in Einzel- oder Teamarbeit
Seminar (S)	Studierende erschließen das Wissen selbstständig im Dialog mit den Lehrenden, zum Beispiel im Rahmen von Referaten, Hausarbeiten, Vorträgen und Diskussionen

Tabelle 1.2: Lehrformen

1.3 Weitere Informationen

Das geplante Lehrangebot wird regelmäßig in einem *2-Jahresplan* auf den Webseiten der Fakultät veröffentlicht, um Studierenden eine möglichst hohe Planbarkeit ihres Studiums zu ermöglichen. Das tatsächliche Angebot in einem Semester kann in Ausnahmefällen aber von diesem Plan abweichen. Weitere Informationen entnehmen Sie bitte der für Sie gültigen *Prüfungsordnung* und dem *Leitfaden*, die Sie auf der Webseite der Fakultät finden.

2 Modulbeschreibungen

2.1 Grundlagenteil / Kompetenzteil Software Engineering

2.1.1 Architekturen moderner Informationssysteme (M)

Modulname	Architekturen moderner Informationssysteme (M) / Architectures of Modern Information Systems
Modulverantwortliche(r)	B. Müller
Lehrveranstaltung	Architekturen moderner Informationssysteme (M) / Architectures of Modern Information Systems
Dauer / Häufigkeit	1 Sem. / Unregelmäßig
Lehrsprache	Deutsch
Prüfungsform/-dauer	K1,5h
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF KT: SE) • M.Sc. Informatik (PO18) (WPF KT: SE)
Empf. Voraussetzungen	Weiterführende Kenntnisse der Software-Entwicklung und Erfahrung in der Implementierung von Informationssystemen sollten vorhanden sein.
Lehr- und Lernformen	4SU

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

- verschiedene Vorgehensweisen zur Architekturentwicklung benennen und beurteilen
- typische Architekturaspekte beurteilen und einschätzen
- Architekturalternativen vergleichen
- Architektur-Patterns und Verteilungs-Patterns benennen und beurteilen
- Architekturentwürfe erstellen, beurteilen, vergleichen und durchführen

Lehrinhalte

- Was ist Architektur?
- Unternehmens-Pattern
- Verteilungs-Patterns
- Domain Driven Design
- Monolithische vs. Microservice-basierte Architekturen
- API Design

Literatur

- Fowler, M. (2002): *Patterns of Enterprise Application Architecture*. Pearson.
- Joshi, U. (2023): *Patterns of Distributed Systems*. Addison Wesley.
- Lilienthal, C. (2019): *Langlebige Software-Architekturen* (3. Aufl.). dpunkt.
- Starke, G. (2020): *Effektive Software-Architekturen* (9. Aufl.). Hanser.
- Zörner, S. (2021): *Software-Architekturen dokumentieren und kommunizieren* (3. Aufl.). Hanser.
- Webber, J., Parastatidis, S., Robinson, I. (2010): *REST in Practice - Hypermedia and Systems Architecture*. O'Reilly.
- Vernon, V., Tomasz, J. (2021): *Strategic Monoliths and Microservices*. Addison-Wesley.
- Richardson, C. (2018): *Microservices Patterns*. Manning.
- Bruce, M., Pereira, P. A. (2018): *Microservices in Action*. Manning.

Versionsnummer: 1 (21.02.2024)

2.1.2 Cloud Native Computing (M)

Modulname	Cloud Native Computing (M) / Cloud Native Computing
Modulverantwortliche(r)	H. Grönniger
Lehrveranstaltung	Cloud Native Computing (M) / Cloud Native Computing
Dauer / Häufigkeit	1 Sem. / Unregelmäßig
Lehrsprache	Deutsch
Prüfungsform/-dauer	EA(30%)+K1,5h(70%)
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF KT: SE) • M.Sc. Informatik (PO18) (WPF KT: SE)
Empf. Voraussetzungen	-
Lehr- und Lernformen	2V + 1Ü + 1L

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

- Definitionen und Konzepte zum Thema Cloud Computing, Cloud Native Computing, DevOps, Telemetrie, Verteilte Anwendungen verstehen und bewerten
- Cloud und on-premises gegenüberstellen und bewerten
- eine Anwendung unter Berücksichtigung von Cloud Computing / Verteilung entwerfen und entwickeln, in Betrieb nehmen und überwachen

Lehrinhalte

- Clouddefinitionen
- Ökonomie
- Infrastructure as a Service
- DevOps
- Branching- und Release-Strategien
- CI/CD Automatisierung
- Umgebungsmanagement
- Virtualisierung, Containermanagement und Docker
- Verteilung, Skalierung
- Everything as code / immutable architecture / IaC
- Function as a Service
- Beobachtbarkeit: Telemetrie: Monitoring, Logging

Literatur

- Kratzke, N. (2023): *Cloud-native Computing - Software Engineering von Diensten und Applikationen für die Cloud* (2. Aufl.). Hanser.

2.1.3 Effiziente Algorithmen (M)

Modulname	Effiziente Algorithmen (M) / Efficient Algorithms
Modulverantwortliche(r)	J. Weimar
Lehrveranstaltung	Effiziente Algorithmen (M) / Efficient Algorithms
Dauer / Häufigkeit	1 Sem. / Unregelmäßig
Lehrsprache	Deutsch
Prüfungsform/-dauer	K1,5h/M/PA
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF KT: SE) • M.Sc. Informatik (PO18) (WPF KT: SE)
Empf. Voraussetzungen	Kenntnisse, wie sie in <i>Algorithmen und Datenstrukturen</i> sowie <i>Theoretische Informatik</i> im Bachelor Informatik vermittelt werden.
Lehr- und Lernformen	4SU

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

- effiziente Algorithmen aus verschiedenen Teilbereichen beschreiben
- die Komplexität eines Algorithmus wissenschaftlich fundiert abschätzen
- die Begriffe der Komplexitätstheorie wiedergeben und das N vs. NP-Problem erläutern
- selbst mehrere Algorithmen umsetzen und die Effizienz in der Praxis testen
- algorithmische Problemlösungen selbstständig erarbeiten

Lehrinhalte

- Wiederholung der Grundlagen von Algorithmen, deren Effizienz, Komplexität und Berechenbarkeit, dynamischen Datenstrukturen
- Mindestens zwei Gebiete aus den folgenden:
 - Algorithmen zu Zeichenketten: effiziente Suche, Vergleich, Suffix-Trees
 - geometrische Algorithmen: Konvexe Hülle, Voronoi-Diagramm, k-nearest Neighbors
 - Algorithmen zur Erfüllbarkeit aussagenlogischer Formeln: Horn-Formeln, Resolution, Heuristiken, SAT-solver

Literatur

- Ottmann, T., Widmayer, P. (2017): *Algorithmen und Datenstrukturen* (6. Aufl.). Springer.
- Apostolico, A., Farach-Colton, M., Galil, Z., Muthukrishnan, S. (2016): 40 years of suffix trees. *Communications of the ACM* 59(4), S. 66–73.

Weitere Literatur wird in der Lehrveranstaltung bekanntgegeben.

Versionsnummer: 1 (21.02.2024)

2.1.4 Entwicklung großer Anwendungssysteme (M)

Modulname	Entwicklung großer Anwendungssysteme (M) / Development of large Application Systems
Modulverantwortliche(r)	B. Müller
Lehrveranstaltung	Entwicklung großer Anwendungssysteme (M) / Development of large Application Systems
Dauer / Häufigkeit	1 Sem. / Unregelmäßig
Lehrsprache	Deutsch
Prüfungsform/-dauer	K1,5h
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF KT: SE) • M.Sc. Informatik (PO18) (WPF KT: SE)
Empf. Voraussetzungen	Grundlegende Kenntnisse der Software-Entwicklung und Erfahrung mit der Implementierung kleinerer Systeme sollten vorhanden sein.
Lehr- und Lernformen	3V + 1Ü

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

- die Problemstellungen großer Anwendungssysteme darstellen und einordnen
- Architekturalternativen bei der Umsetzung großer Anwendungssysteme bewerten
- große Anwendungssysteme erstellen und bewerten
- verschiedene Jakarta-EE-APIs als Problemlösungskomponenten verschiedener Anforderungen benennen und einsetzen

Lehrinhalte

Ausgesuchte Bereiche der Jakarta-EE-Spezifikation

- Grundlegende Jakarta APIs
- Transactions
- Asynchrone RESTful Web Services
- WebSockets
- Messaging
- Concurrency
- JSON-P, JSON-B
- Security

Literatur

- Starke, G. (2018): *Effektive Software-Architekturen*. Hanser.
- Richards, M., Ford, N. (2020): *Fundamentals of Software Architecture*. O'Reilly.
- Eclipse Foundation (2024): *Jakarta EE*. Technischer Bericht, <https://jakarta.ee/specifications>.

Versionsnummer: 1 (21.02.2024)

2.1.5 Formale Methoden (M)

Modulname	Formale Methoden (M) / Formal Methods
Modulverantwortliche(r)	M. Huhn
Lehrveranstaltung	Formale Methoden (M) / Formal Methods
Dauer / Häufigkeit	1 Sem. / Unregelmäßig
Lehrsprache	Deutsch
Prüfungsform/-dauer	K1,5h(50%)+PA(50%)
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF KT: SE) • M.Sc. Informatik (PO18) (WPF KT: SE)
Empf. Voraussetzungen	-
Lehr- und Lernformen	2V + 1Ü + 1L

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

- digitale Systeme und Systemeigenschaften analysieren und formal modellieren
- grundlegende Algorithmen beschreiben, um Systemeigenschaften automatisch formal zu analysieren und zu verifizieren
- eine werkzeuggestützte formale Verifikation für kleinere Beispiele durchführen
- die Möglichkeiten und Grenzen des Einsatzes formaler Methoden beurteilen

Lehrinhalte

Modellierung und Analyse des Systemsverhaltens mit formalen Verifikationstechniken, insb. Model Checking, Verwendung von Verifikationswerkzeugen typischer Szenarien in Software-, Hardware- und cyberphysikalischen Systemen.

Literatur

- Baier, C., Katoen, J.-P. (2008): *Principles of Model Checking*. MIT Press.

2.1.6 IT-Management (M)

Modulname	IT-Management (M) / IT Management
Modulverantwortliche(r)	H. Grönniger
Lehrveranstaltung	IT-Management (M) / IT Management
Dauer / Häufigkeit	1 Sem. / Unregelmäßig
Lehrsprache	Deutsch
Prüfungsform/-dauer	K1,5h/PA
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF KT: SE) • M.Sc. Informatik (PO18) (WPF KT: SE)
Empf. Voraussetzungen	-
Lehr- und Lernformen	2V + 1Ü + 1L

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

- Themen des IT-Management verstehen und einordnen
- Struktur und Abläufe einer Unternehmens-IT analysieren und bewerten
- IT-Management Prozesse aus den Bereichen IT-Strategie, IT-Service, Management, Enterprise Architecture, IT-Projektmanagement anwenden
- Infrastrukturmanagement am Beispiel Cloud Computing verstehen und bewerten

Lehrinhalte

- Management-Perspektive und technologische Sicht
- Aufbau und Einbettung der IT in Unternehmen
- Übersicht über die Aufgabengebiete der IT und des IT-Management
- IT-Infrastrukturen mit praktischen Teilen im Bereich Cloud / Infrastructure-as-a-Service
- IT-Strategie: Entwicklung von IT-Strategien im Unternehmen
- IT-Services und IT-Service Management: wie wird der reibungslose Ablauf des operativen IT-Betriebs organisiert
- Enterprise Architecture Management: wie werden Geschäftsprozesse und Anwendungslandschaft in Einklang gebracht
- IT-Projekte: bringen Veränderung in die IT-Landschaft und müssen je nach Art der Aufgabe strukturiert und im Unternehmen verankert werden

Literatur

Literatur wird in der Lehrveranstaltung bekanntgegeben.

2.1.7 Modellgetriebene Softwareentwicklung (M)

Modulname	Modellgetriebene Softwareentwicklung (M) / Model-Driven Software Development
Modulverantwortliche(r)	M. Huhn
Lehrveranstaltung	Modellgetriebene Softwareentwicklung (M) / Model-Driven Software Development
Dauer / Häufigkeit	1 Sem. / Unregelmäßig
Lehrsprache	Deutsch
Prüfungsform/-dauer	EA(50%)+K1,5h(50%)
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF KT: SE) • M.Sc. Informatik (PO18) (WPF KT: SE)
Empf. Voraussetzungen	-
Lehr- und Lernformen	2V + 1Ü + 1L

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

- das Paradigma und die Konzepte der modellgetriebenen Softwareentwicklung, und aktuelle Ausprägungen in Low-Code-Frameworks beschreiben
- mögliche Anwendungsbereiche und Grenzen modellgetriebener Softwareentwicklung erkennen und geeignete Ansätze auswählen
- mit den in der Vorlesung vorgestellten Werkzeugen konkrete MDSD-Fallstudien wie etwa die Entwicklung einer Mini-DSL Entwicklung incl. Modelltransformation oder Codegenerierung durchführen

Lehrinhalte

- Modellgetriebene Entwicklungsprozesse
- OMG MDA vs. MDSD
- UML, UML Profile, AADL
- Eclipse Modeling Framework: Modell-Transformationen
- DSL Engineering
- Codegenerierung als Model2Text Transformation
- MDSD-Werkzeuge
- Modell-Analyse
- Low-Code

Literatur

Literatur wird in der Lehrveranstaltung bekanntgegeben.

2.1.8 User Interfaces für Mobile Systeme (M)

Modulname	User Interfaces für Mobile Systeme (M) / User Interfaces for Mobile Systems
Modulverantwortliche(r)	J. Weimar
Lehrveranstaltung	User Interfaces für Mobile Systeme (M) / User Interfaces for Mobile Systems
Dauer / Häufigkeit	1 Sem. / Unregelmäßig
Lehrsprache	Deutsch
Prüfungsform/-dauer	(R(15%)+PA(35%)+K1,5h(50%))/(K1,5h(50%)+PA(50%))
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF KT: SE) • M.Sc. Informatik (PO18) (WPF KT: SE)
Empf. Voraussetzungen	-
Lehr- und Lernformen	4SU

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

- mobile Softwaresysteme theoretisch fundiert und mit systematischen Ansätzen benutzergerecht und gebrauchstauglich gestalten
- zentrale Begriffe, gesetzliche Grundlagen und Normen benennen
- die besonderen Eigenschaften mobiler Eingabegeräte beschreiben und können Anwendungen spezifisch darauf zuschneiden
- benutzerzentrierte Vorgehensmodelle der Software-Ergonomie im Software-Entwicklungsprozess für mobile Systeme erläutern
- Methoden zur nutzerbezogenen Anforderungsanalyse verwenden

Lehrinhalte

- Benutzeroberflächen gängiger mobiler Geräte
- Gestaltungsrichtlinien für iOS und Android
- innovative Bedienkonzepte bei mobilen Geräten
- Einsatz von verschiedensten Sensoren
- Nutzung von Kontext
- Oberflächenentwicklung für Android und/oder iOS

Literatur

- <https://developer.android.com>
- <https://developer.apple.com>

2.1.9 Werkzeuge der Java-Plattform (M)

Modulname	Werkzeuge der Java-Plattform (M) / Java Platform Tools
Modulverantwortliche(r)	B. Müller
Lehrveranstaltung	Werkzeuge der Java-Plattform (M) / Java Platform Tools
Dauer / Häufigkeit	1 Sem. / Unregelmäßig
Lehrsprache	Deutsch
Prüfungsform/-dauer	K1,5h
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF KT: SE) • M.Sc. Informatik (PO18) (WPF KT: SE)
Empf. Voraussetzungen	Grundlegende Kenntnisse der Software-Entwicklung, der Programmiersprache Java und der Laufzeitumgebung JVM sollten vorhanden sein.
Lehr- und Lernformen	3V + 1Ü

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

- den Aufbau der Java-Virtual-Machine erläutern
- reflexive Konzepte von Programmiersprachen auf der Basis von Java einsetzen
- die Konzepte Class-Loader, Class-Transformer und Agenten praxisnah einsetzen
- verschiedene Aspekte der Java-Laufzeitumgebung wie Management-Beans, Dynamische Proxies, Garbage-Collection und Threading beurteilen und für konkrete Anwendungsfälle umsetzen
- verschiedene Übersetzungsstrategien beurteilen und einsetzen

Lehrinhalte

- Reflection
- Class-Loader
- Class-Transformer und Agenten
- Management-Beans
- Dynamic Proxy
- GraalVM und native Images
- Speicherverwaltung und Garbage-Collection
- Threading und Synchronisation
- JIT / AOT / Tiered Compilation

Literatur

- Oaks, S. (2020): *Java Performance: In-depth Advice for Tuning and Programming Java 8, 11, and Beyond* (2. Aufl.). O'Reilly.
- Evans, B. J., Gough, J., Newland, C. (2018): *Optimizing Java - Practical Techniques for Improving JVM Application Performance*. O'Reilly.
- Hunt, C. (2011): *Java Performance*. Addison-Wesley.

2.2 Grundlagenteil / Kompetenzteil Data Science

2.2.1 Statistische Methoden (M)

Modulname	Statistische Methoden (M) / Statistical Methods
Modulverantwortliche(r)	F. Klawonn
Lehrveranstaltung	Statistische Methoden (M) / Statistical Methods
Dauer / Häufigkeit	1 Sem. / Unregelmäßig
Lehrsprache	Deutsch
Prüfungsform/-dauer	
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF KT: DS) • M.Sc. Informatik (PO18) (WPF KT: IE)
Empf. Voraussetzungen	Grundkenntnisse der Statistik/Wahrscheinlichkeitsrechnung sollten vorhanden sein.
Lehr- und Lernformen	2V + 1Ü + 1L

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

- erkennen, wann welche statistische Verfahren sinnvoll anzuwenden sind
- erkennen, ob statistische Verfahren korrekt angewandt und die Interpretation der Ergebnisse korrekt vorgenommen wurden
- Beispieldaten mit Hinterfragung des experimentellen Designs auswerten
- die Modellierungsannahmen der inferenziellen Statistik wiedergeben

Lehrinhalte

- Zufallsvariablen und statistische Modellierung
- Punktschätzer
- Konfidenzintervalle
- Hypothesentestes
- Korrelation und Regressionsanalyse
- Robuste Statistik
- Bayessche Statistik
- Logistische Regression
- Überlebenszeit-/Time-to-Event-Analyse

Literatur

- Dalgaard, P. (2008): *Introductory Statistics with R* (2. Aufl.). Springer.
- Field, A., Miles, J., Field, Z. (2013): *Discovering Statistics Using R*. Sage.

2.2.2 Heuristische Suche (M)

Modulname	Heuristische Suche (M) / Heuristic Search
Modulverantwortliche(r)	K. Gutenschwager
Lehrveranstaltung	Heuristische Suche (M) / Heuristic Search
Dauer / Häufigkeit	1 Sem. / Unregelmäßig
Lehrsprache	Deutsch
Prüfungsform/-dauer	EA(40%)+K1,5h(60%)
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF KT: DS) • M.Sc. Informatik (PO18) (WPF KT: IE)
Empf. Voraussetzungen	Grundlagen im Bereich Algorithmen und Datenstrukturen sollten bekannt sein.
Lehr- und Lernformen	2V + 2L

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

- die Grenzen exakter mathematischer Verfahren für np-schwere Probleme benennen
- die unterschiedlichen Lösungsstrategien und Verfahren aus dem Bereich der Meta-Heuristiken erklären
- Probleme selbständig analysieren und formal formulieren
- allgemeine Lösungsverfahren für neue Problemstellungen eigenständig adaptieren und implementieren
- Experimente selbständig konzipieren, durchführen und die Ergebnisse einem Fachpublikum präsentieren

Lehrinhalte

- Einführung in Optimierungsprobleme
 - Formulierung
 - Lösbarkeit
- Constraint-Satisfaction-Problem
 - Anwendungsbereiche und Problemformulierungen
 - Lösungsansätze (AC-3-Algorithmus, Backtracking, Min-Conflict-Heuristik)
- Grundlegende Lösungsstrategien
 - Eröffnungs- vs. Verbesserungsverfahren
 - Lokale Suche und evolutionäre Ansätze
- Meta-Heuristiken
 - Tabu Search
 - Simulated Annealing
 - Ant Search
 - Genetische Algorithmen
 - Scatter Search
 - Variable Neighborhood Search
- Ausgewählte Problemstellungen und Anwendungsbeispiele

Literatur

Literatur wird in der Lehrveranstaltung bekanntgegeben.

2.2.3 Neuronale Netze und Deep Learning (M)

Modulname	Neuronale Netze und Deep Learning (M) / Neural Networks and Deep Learning
Modulverantwortliche(r)	C. Meyer
Lehrveranstaltung	Neuronale Netze und Deep Learning (M) / Neural Networks and Deep Learning
Dauer / Häufigkeit	1 Sem. / Unregelmäßig
Lehrsprache	Englisch
Prüfungsform/-dauer	K1,5h
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF KT: DS) • M.Sc. Informatik (PO18) (WPF KT: IE)
Empf. Voraussetzungen	<ul style="list-style-type: none"> • Kenntnisse Mathematische Grundlagen in Algebra, Analysis und Optimierung sind hilfreich. • Programmierkenntnisse in Python und Kenntnisse im Umgang mit Softwarebibliotheken (Python) sind vorteilhaft. • großes Interesse an neuronalen Netzen und Deep Learning sowie hohe Motivation erforderlich. • überdurchschnittliche Fähigkeiten zu konzeptionellem und analytischen Denken vorteilhaft.
Lehr- und Lernformen	2V + 1L + 1U

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

- Konzepte, Algorithmen, Architekturen und Anwendungen von neuronalen Netzen und Deep Learning grundlegend überblicken
- die Möglichkeiten und Grenzen der Technologie verstehen
- geeignete einfache neuronale Netze und Deep-Learning-Techniken auf neue Probleme anwenden (inklusive Training der Netzwerke)
- die Ergebnisse von Experimenten mit neuronalen Netzen und Deep Learning analysieren und darstellen
- Möglichkeiten zur Verbesserung der Performanz skizzieren und sie umsetzen

Lehrinhalte

- Biologische Grundlagen (Neuronen und Netzwerke)
- Künstliche Neuronenmodelle
- Architekturtypen künstlicher neuronaler Netze
- Vorwärtsgerichtete neuronale Netze: Das Mehrschicht-Perzeptron
- Lernen in neuronalen Netzen und der Backpropagation-Algorithmus
- Deep Learning: Motivation und Konzepte
- Convolutional neural networks
- Weitere Themen wie LSTM, Autoencoder, GAN

Literatur

- Goodfellow, I., et al. (2016): *Deep Learning*. MIT Press.
- Nielsen, M. A., (2015): *Neural Networks and Deep Learning*. Determination Press, <http://neuralnetworksanddeeplearning.com/>.

Weitere Literatur wird in der Lehrveranstaltung bekanntgegeben.

2.2.4 Automatische Sprachverarbeitung (M)

Modulname	Automatische Sprachverarbeitung (M) / Automatic Speech Processing
Modulverantwortliche(r)	C. Meyer
Lehrveranstaltung	Automatische Sprachverarbeitung (M) / Automatic Speech Processing
Dauer / Häufigkeit	1 Sem. / Unregelmäßig
Lehrsprache	Deutsch
Prüfungsform/-dauer	K1,5h
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF KT: DS) • M.Sc. Informatik (PO18) (WPF KT: IE)
Empf. Voraussetzungen	Kenntnisse mathematischer Grundlagen in Algebra, Analysis sowie Wahrscheinlichkeitstheorie und Statistik sollten vorhanden sein.
Lehr- und Lernformen	2V + 1L + 1Ü

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende

- wichtige Anwendungen der automatischen Sprachverarbeitung hinsichtlich ihrer Anforderungen und Randbedingungen unterscheiden
- grundlegende Algorithmen der automatischen Sprachverarbeitung (Schwerpunkt Spracherkennung) erläutern, ihre Vor- und Nachteile einschätzen und ihre Eignung in konkreten Anwendungsszenarien beurteilen
- aktuelle Möglichkeiten und Grenzen der Technologie einschätzen
- Ergebnisse von Spracherkennungs-Experimenten analysieren und darstellen

Lehrinhalte

- Anwendungen, Systeme und Architekturen
- Merkmalsextraktion
- Akustische Modellierung
- Training und Adaptationsmethoden
- Sprachmodelle
- Suche
- Ausgewählte Methoden des Sprachverstehens
- Dialogsysteme

Literatur

- Rabiner, L., Juang, B.-H. (1993): *Fundamentals of Speech Recognition*. Prentice Hall.
 - Huang, X., Acero, A., Hon H.-W. (2001): *Spoken Language Processing*. Prentice Hall.
 - Jelinek, F. (1998): *Statistical Methods for Speech Recognition*. MIT Press.
 - Jurafsky, D., Martin, J. H. (2008): *Speech and Language Processing* (2. Aufl.). Prentice Hall.
- Weitere Literatur wird in der Lehrveranstaltung bekanntgegeben.

2.2.5 Datenbanktechnologien (M)

Modulname	Datenbanktechnologien (M) / Database Technologies
Modulverantwortliche(r)	F. Höppner
Lehrveranstaltung	Datenbanktechnologien (M) / Database Technologies
Dauer / Häufigkeit	1 Sem. / Unregelmäßig
Lehrsprache	Deutsch
Prüfungsform/-dauer	EA(10%)+K1,5h(90%)
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF KT: DS) • M.Sc. Informatik (PO18) (WPF KT: IE)
Empf. Voraussetzungen	Kenntnisse zur Abfragesprache SQL und zur relationalen Datenmodellierung sollten vorhanden sein.
Lehr- und Lernformen	3SU + 1Ü

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

- grundlegende Technologie zur Sicherung der ACID-Eigenschaften (z.B. Recovery) von Transaktionen einordnen und teilweise umsetzen
- die Evaluation von Anfrageplänen um Operatoren selbstständig erweitern und gegebene Abfragen bewerten
- für unterschiedliche Anforderungsprofile analysieren, welche Technologien aus der SQL/NoSQL-Welt angemessen ist
- Datenbanktechnologien bzgl. ihrer Konsistenzmodelle bewerten
- eigene Lösungen zu Aufgaben mit Kommilitonen fachlich fundiert diskutieren

Lehrinhalte

- Datenorganisation (Speicherung von Tupeln, Relationen, Indizes)
- Anfrage-Optimierung in relationalen DBMS (Anfrageplan, Optimierung)
- Recovery nach Ausfall
- Synchronisation von Transaktionen (Serialisierung, Scheduler)
- Verteilte Datenbanken (Partitionierung, Auswirkung auf Optimierung und Synchronisation)
- Transition zu NoSQL (CAP Theorem, NoSQL-DB-Typen)
- Multiversion Concurrency Control, Eventually-Consistent Data Types
- Verarbeitung großer Datenmengen (Hadoop, Map/Reduce, Hive, HBase)
- In-Memory-Datenbanken

Literatur

- Kemper, A., Eickler, A. (2015): *Datenbanksysteme* (10. Aufl.). De Gruyter.
- Plattner, H. (2013): *Lehrbuch In-Memory Data Management*. Springer.
- White, T. (2015): *Hadoop: The Definitive Guide* (4. Aufl.). O'Reilly.
- Wiese, L. (2015): *Advanced Data Management*. de Gruyter.

2.2.6 Big Data (M)

Modulname	Big Data (M) / Big Data
Modulverantwortliche(r)	F. Höppner
Lehrveranstaltung	Big Data (M) / Big Data
Dauer / Häufigkeit	1 Sem. / Unregelmäßig
Lehrsprache	Deutsch
Prüfungsform/-dauer	EA(40%)+K1,5h(60%)
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF KT: DS) • M.Sc. Informatik (PO18) (WPF KT: IE)
Empf. Voraussetzungen	Umsetzung der Aufgaben erfordern Java-Kenntnisse.
Lehr- und Lernformen	3SU + 1Ü

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

- spezifische Probleme und Engpässe für algorithmische Lösungsansätze einordnen und das Spektrum an Lösungsansätzen verstehen
- technische Lösungen auf einem Hadoop-Cluster umsetzen
- Hadoop/Spark-Lösungen theoretisch und praktisch bzgl. ihrer Performance analysieren und bewerten
- die Methoden und Techniken auf neue Problemstellungen übertragen, eine eigene effiziente Lösung (in mehreren Stufen) selbständig erarbeiten und mit den anderen Lösungen im Vergleich evaluieren

Lehrinhalte

- Aufbau der verteilten Infrastruktur Apache Hadoop (HDFS, Map/Reduce, Spark), Abläufe, Engpässe
- verteilte Informationsverarbeitung mit Hadoop Map/Reduce in Java
- Map/Reduce Design-Patterns
- Effekte von Skew, theoretische Komplexitätsabschätzungen
- Aufbau und Funktionsweise Apache Spark, Unterschiede Map/Reduce
- Umsetzungen mit Spark, Resilient Distributed Datasets, Transformationen und Jobs
- Ausgewählte Problemstellungen/Lösungstechniken aus Bereichen Datenanalyse und Inf-Retrieval (ml-lib, Bloom-Filter, Locality Sensitive Hashing, Recommendation Systems, Clusteranalyse, etc.)

Literatur

- White, T. (2015): *Hadoop: The Definitive Guide* (4. Aufl.). O'Reilly.
- Damji, J. S., Wenig, B., Das, T., Lee, D. (2020): *Learning Spark: Lightning-Fast Big Data Analysis* (2. Aufl.). O'Reilly.
- Odersky, M., Spoon, L., Venners, B., Sommers, F. (2021): *Programming in Scala* (5. Aufl.). artima.

Versionsnummer: 1 (21.02.2024)

2.2.7 Angewandte Kryptographie (M)

Modulname	Angewandte Kryptographie (M) / Applied Cryptography
Modulverantwortliche(r)	I. Schiering
Lehrveranstaltung	Angewandte Kryptographie (M) / Applied Cryptography
Dauer / Häufigkeit	1 Sem. / Unregelmäßig
Lehrsprache	Englisch
Prüfungsform/-dauer	PF(20%)+K1,5h(80%)
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF KT: DS) • M.Sc. Informatik (PO18) (WPF KT: IE)
Empf. Voraussetzungen	Kenntnisse diskreter Mathematik, insbesondere Zahlentheorie, Gruppen, Körper sollten vorhanden sein.
Lehr- und Lernformen	2V + 1Ü + 1P

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

- kryptographische Verfahren und die mathematischen Grundlagen dazu beschreiben
- Sicherheitsanforderungen von Kommunikation analysieren
- kryptographische Verfahren und Protokolle dazu auswählen bzw. potentielle Schwachstellen herausarbeiten und bewerten
- sich neue Aspekte in diesem Zusammenhang eigenständig erarbeiten und umsetzen
- im Rahmen der Projekte Forschungsmethodiken anhand konkreter Fragestellungen des Security Engineering und Privacy Engineering erproben

Lehrinhalte

- Stream Ciphers / Block Ciphers
- Symmetric Encryption (AES)
- Modes of Operation (ECB, CBC, OFB, ...)
- Asymmetric Ciphers (RSA, Diffie-Hellman, ECC, ...)
- Anwendungen: Auswahl aus Digitale Signaturen, Cryptographic Hash Functions, Message Authentication Codes, Key Management
- Fortgeschrittene Themen: Auswahl aus Zero-Knowledge Proofs, Post-Quantum Cryptography, Blockchain, Homomorphic Encryption, etc.
- Innerhalb eines begleitenden Projekts setzen sich Studierende wissenschaftlich mit Aspekten des Security Engineering und Privacy Engineering auseinander.

Literatur

- Paar, C., Pelzl, J., Güneysu, T. (2024): *Understanding Cryptography: From Established Symmetric and Asymmetric Ciphers to Post-Quantum Algorithms*. Springer.
- Bernstein, D. J., Buchmann, J., et al. (2008): *Post-Quantum Cryptography*. Springer.
- Menezes, A. J., van Oorschot, P. C., Vanstone, S.A. (1996): *Handbook of applied cryptography*. CRC Press.

Weitere Literatur wird in der Lehrveranstaltung bekanntgegeben.

2.2.8 Machine Learning (M)

Modulname	Machine Learning (M) / Machine Learning
Modulverantwortliche(r)	D. Lehmann
Lehrveranstaltung	Machine Learning (M) / Machine Learning
Dauer / Häufigkeit	1 Sem. / Unregelmäßig
Lehrsprache	Deutsch
Prüfungsform/-dauer	K1,5h
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF KT: DS) • M.Sc. Informatik (PO18) (WPF KT: IE)
Empf. Voraussetzungen	-
Lehr- und Lernformen	4SU

Qualifikationsziele

Nach erfolgreichem Abschluss können Studierende:

- grundlegenden Begriffe und Verfahren des maschinellen Lernens kompetent nutzen
- komplexe interdisziplinäre Analyseprobleme in geeigneter Form formalisieren und passende Verfahren/Modelle zur Lösung dieser Probleme einsetzen
- Modelle zur Analyse großer Datenmengen selbstständig erstellen/trainieren und wiss. beurteilen
- die Güte eines (multidimensionalen oder multivariaten) Datensatzes einschätzen
- maschinelles Lernen zur Assoziationsanalyse, Clustering, Klassifikation, Regression, Anomalieerkennung und Zeitreihenanalyse anwenden, bewerten und evaluieren.
- die Güte berechneter Modelle beurteilen und diese auf neue unvertraute Problemstellungen übertragen.

Lehrinhalte

Behandelt werden u.a. folgende Themen:

- Grundbegriffe
- Ablauf von Data Science Projekten
- Erkunden und kennelernen von Daten
- Assoziationsanalysen
- Clusteralgorithmen (k-Means, EM, DBSCAN, Single Linkage)
- Klassifikation (Nearest Neighbor, Entscheidungsbäume + Pruning Verfahren, Random Forest, Logistic Regression, Naive Bayes, SVM, Neuronale Netze)
- Inliers-/Outlieranalyse
- Regression höherer Ordnung
- Zeitreihenanalyse u.a. mit ARIMA
- Gaussian Mixture Models (GMM)
- Hidden Markov Model (HMM)
- Conditional Random Fields
- Evaluationsmethoden für gelernte Modelle
- Nutzung der genannten Verfahren mit Bibliotheken für die Programmiersprache Python

Literatur

Literatur wird in der Lehrveranstaltung bekanntgegeben.

2.2.9 Stream Processing (M)

Modulname	Stream Processing (M) / Stream Processing
Modulverantwortliche(r)	D. Lehmann
Lehrveranstaltung	Stream Processing (M) / Stream Processing
Dauer / Häufigkeit	1 Sem. / Unregelmäßig
Lehrsprache	Deutsch
Prüfungsform/-dauer	K1,5h
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF KT: DS) • M.Sc. Informatik (PO18) (WPF KT: IE)
Empf. Voraussetzungen	-
Lehr- und Lernformen	4SU

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

- Echtzeitdatenverarbeitung in verschiedenen Anwendungsbereichen bewerten und einsetzen
- komplexe Anforderungen moderner datengetriebener Applikationen bewältigen
- Einsatzgebiete und Technologien der Echtzeit-Verarbeitung und Analyse von Datenströmen überblicken
- Technologien selbstbewusst auswählen und erste Lösungen umsetzen
- Methoden des Stream Processing und der Erstellung von Modellen (online) beschreiben
- Probleme und Anwendungsgebiete des Stream Processing selbstständig bearbeiten, sowie prominenter Use Cases diesbezüglich skizzieren
- Lösungen, Modelle und Methoden für einfache und mittlere interdisziplinäre Problemstellungen des Stream Processing eigenständig bzw. in kleineren Teams erfolgreich erarbeiten
- Lösungen und Modelle eigenständig technisch umsetzen, evaluieren und in der gewünschten Applikation integrieren

Lehrinhalte

- Einführung in das Stream Processing:
 - Definitionen, Konzepte und Bedeutung
 - Unterschiede zwischen Batch- und Stream Processing
- Stream-Modelle und -Architekturen:
 - Batch-, Microbatch- und Continuous-Processing-Modelle
 - Event-Time vs. Processing-Time, Fensterung und Zeitfenster
- Stream-Processing-Frameworks wie, z. B.:
 - Apache Kafka: Streaming-Plattform
 - Apache Flink: Echtzeitdatenverarbeitung und -analyse
- Datenstrom-Operatoren und -Transformationen:
 - Filterung, Transformation, Aggregation
 - Zustandsmanagement und Caching
- Skalierbare Architekturen für Stream Processing:
 - Horizontale Skalierung, Lastverteilung und Ausfallsicherheit
 - Verarbeitung großer Datenströme verteilter Umgebungen
 - Erstellung von datengetriebenen Modellen aus Streams/Online
 - Online-Algorithmen zur Verarbeitung von Daten
- Anwendungen und Fallstudien:
 - Echtzeit-Analytik für Unternehmensdaten
 - IoT-Anwendungen und Sensordatenverarbeitung
- Hands-on-Übungen und Projekte:
 - Implementierung von Stream-Processing-Szenarien
 - Verwendung von Frameworks zur Umsetzung von Echtzeitverarbeitungsaufgaben

Literatur

Literatur wird in der Lehrveranstaltung bekanntgegeben.

Versionsnummer: 1 (21.02.2024)

2.3 Grundlagenteil / Kompetenzteil Systems & Computer Engineering

2.3.1 Automotive Systems (M)

Modulname	Automotive Systems (M) / Automotive Systems
Modulverantwortliche(r)	G. Bikker
Lehrveranstaltung	Automotive Systems (M) / Automotive Systems
Dauer / Häufigkeit	1 Sem. / Unregelmäßig
Lehrsprache	Deutsch
Prüfungsform/-dauer	EA(30%)+K1h(70%)
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF KT: SCE) • M.Sc. Informatik (PO18) (WPF KT: SCE)
Empf. Voraussetzungen	Programmierkenntnisse, Kenntnisse objektorientierter Programmierung oder Modellierung oder Softwareengineering sollten vorhanden sein.
Lehr- und Lernformen	2V + 1Ü + 1L

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

-die wesentlichen Begriffe, Methoden sowie Anwendungsbereiche benennen

- unter Berücksichtigung von funktionaler Sicherheit fahrzeugrelevante Systeme entwerfen
- selbst entwickelte Assistenzsysteme simulieren und haben sich mit der Integration von Umfeldsensorik und Umwelteinflüssen beschäftigt
- Konzepte zur Datenfusion, Anwendungen wie Umfeld-Wahrnehmung und V2X (Vehilce-to-Everything Kommunikation) entwerfen

Lehrinhalte

- Grundlagen der Fahrzeugsysteme und der Fahrerassistenzsysteme nach SAE-Standard
- Rahmenbedingungen der Entwicklung und der Entwicklungsprozesse
- Funktionale Sicherheit (Rückverfolgbarkeit, Verifikation und Validierung)
- Virtuelle Integration und Test von FAS
- Sensorik und Aktuatorik für FAS
- Maschinelles Sehen, Datenfusion und Umfeld-Wahrnehmung
- Anwendungen, z.B. V2X Kommunikation und Infrastruktur sowie autonomes Fahren

Literatur

- Reif, K. (2010): *Fahrstabilisierungssysteme und Fahrerassistenzsysteme*. Vieweg+Teubner.
- Maurer, M., et al. (2015): *Autonomes Fahren*. Springer.
- Johanning, V., Mildner, R. (2015): *Car IT kompakt*. Springer.

2.3.2 Autonomous Systems (M)

Modulname	Autonomous Systems (M) / Autonomous Systems
Modulverantwortliche(r)	R. Gerndt
Lehrveranstaltung	Autonomous Systems (M) / Autonomous Systems
Dauer / Häufigkeit	1 Sem. / Sommersemester
Lehrsprache	Englisch
Prüfungsform/-dauer	K1h(50%)+PF(50%)
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF KT: SCE) • M.Sc. Informatik (PO18) (WPF KT: SCE)
Empf. Voraussetzungen	<ul style="list-style-type: none"> • None, but basic knowledge on robots could be advantageous
Lehr- und Lernformen	FC, 2SU + 2L

Qualifikationsziele

After successful participation students

- have a well founded understanding of the field of mobile robotics
- are able to analyse robotic systems and identify typical elements
- are able to select and combine suitable Elements hardware and software) to achieve a required functionality.

Lehrinhalte

- Robot Vision (e.g. image processing, feature extraction)
- Artificial Neural Networks for robotics
- Task execution and manipulation
- Mapping, Path planning, Location, Simultaneous Location and Mapping (SLAM)
- Robot simulation
- Uncertainty and error propagation
- Setting up a robot software stack for Robot Operating System (ROS) mobile robot operation

Literatur

- Corell, N., Hayes, B., Heckmann, C., Roncone, A. (2022): *Introduction to Autonomous Robots: Mechanisms, Sensors, Actuators, and Algorithms*. The MIT Press.
- Dudek, G., Jenkin, M. (2024): *Computational Principles of Mobile Robotics* (3. Aufl.). Cambridge University Press.

Versionsnummer: 1 (21.02.2024)

2.3.3 Robotik (M)

Modulname	Robotik (M) / Robotics
Modulverantwortliche(r)	R. Gerndt
Lehrveranstaltung	Robotik (M) / Robotics
Dauer / Häufigkeit	1 Sem. / Wintersemester
Lehrsprache	Englisch
Prüfungsform/-dauer	K1h(50%)+PF(50%)
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF KT: SCE) • M.Sc. Informatik (PO18) (WPF KT: SCE)
Empf. Voraussetzungen	-
Lehr- und Lernformen	FC, 2SU + 2L

Qualifikationsziele

After successful participation students:

- have an overview of the basics of stationary robots
- can analyse robotics system, identify their components and understand interaction schemes of stationary robots
- design and build stationary robots from components and software modules
- are able to implement basic software modules for stationary robots

Lehrinhalte

- Introduction to robotics and their abstract representation
- Kinematics (forward and inverse)
- Introduction to cobots
- Robot actuators and sensors
- Robot operating system (ROS)
- Grasping
- Practical exercises with cobots

Literatur

- Corell, N., Hayes, B., Heckmann, C., Roncone, A. (2022): *Introduction to Autonomous Robots: Mechanisms, Sensors, Actuators, and Algorithms*. The MIT Press.
- Dudek, G., Jenkin, M. (2024): *Computational Principles of Mobile Robotics* (3. Aufl.). Cambridge University Press.

Versionsnummer: 1 (21.02.2024)

2.3.4 Robuste Systeme (M)

Modulname	Robuste Systeme (M) / Dependability
Modulverantwortliche(r)	C. Führer
Lehrveranstaltung	Robuste Systeme (M) / Dependability
Dauer / Häufigkeit	1 Sem. / Unregelmäßig
Lehrsprache	Deutsch
Prüfungsform/-dauer	PF(30%)+K1h/M(70%)
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF KT: SCE) • M.Sc. Informatik (PO18) (WPF KT: SCE)
Empf. Voraussetzungen	Kenntnisse in Programmieren, Modellierung in UML, Statistik und Software Engineering zum Beispiel aus einem Bachelorstudium in Informatik oder Elektro- und Informationstechnik sollten vorhanden sein.
Lehr- und Lernformen	2V + 1Ü + 1L

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

- einen Überblick über einen normenkonformen Systemlebenszyklus und Entwicklungsprozess geben
- die Konzepte RAMS (Reliability, Availability, Maintainability, Safety) verstehen und voneinander abgrenzen
- Zuverlässigkeitsanalysen durchführen und interpretieren
- eine Gefährdungs- und Risikoanalyse durchführen und daraus SIL/ASIL ableiten
- qualitätssichernde Maßnahmen und Architekturmuster für sichere Systeme beurteilen, auswählen und anwenden
- sich für ein gegebenes Problem Architekturen und Entwicklungsprozesse anhand domänenspezifischer Normen für sichere Systeme wie z.B. EN 5012x oder ISO 26262 unter Berücksichtigung des Safety Integrity Levels (SIL/ASIL) erarbeiten

Lehrinhalte

- Definition der Technischen Zuverlässigkeit (RAMS - Reliability, Availability, Maintenance, Safety) und Robustheit von Systemen.
- Anforderungsmanagement für sicherheitsgerichtete Systeme
- Kennenlernen und Einordnen von relevanten Gesetzen und Normen
- Entwurfsprozesse (Vorgehensmodell wie das V-Modell und Möglichkeiten der modellbasierten Entwicklung)
- Testen von robusten Systemen

Literatur

- Löw, P., Pabst, R., Petry, E. (2010): *Funktionale Sicherheit in der Praxis*. dpunkt.
- Ross, H.-L. (2019): *Funktionale Sicherheit im Automobil* (2. Aufl.). Hanser.
- Meyna, A., Althaus, D., Braasch, A., Plinke, F., Schlummer, M. (2023): *Sicherheit und Zuverlässigkeit technischer Systeme*. Hanser.
- Börcsök, J. (2021): *Funktionale Sicherheit: Grundzüge sicherheitstechnischer Systeme* (5. Aufl.). VDE Verlag.
- Rembold, D. (2024): *Safety Engineering: Das Praxisbuch für funktionale Sicherheit. Sichere und robuste Systeme entwickeln*. Rheinwerk Verlag.
- Birolini, A. (2017): *Reliability Engineering: Theory and Practice* (8. Aufl.). Springer.
- Grundnorm IEC 61508 sowie domänenspezifische Normen wie CENELEC EN 5012x oder ISO 26262 (wird in der Vorlesung bekanntgegeben)

Versionsnummer: 1 (21.02.2024)

2.3.5 Simulation und Verifikation (M)

Modulname	Simulation und Verifikation (M) / Simulation and Verification
Modulverantwortliche(r)	G. Bikker
Lehrveranstaltung	Simulation und Verifikation (M) / Simulation and Verification
Dauer / Häufigkeit	1 Sem. / Unregelmäßig
Lehrsprache	Deutsch
Prüfungsform/-dauer	EA(30%)+K1h(70%)
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF KT: SCE) • M.Sc. Informatik (PO18) (WPF KT: SCE)
Empf. Voraussetzungen	Programmierkenntnisse, Kenntnisse in objektorientierter Programmierung oder Modellierung oder Softwareengineering sollten vorhanden sein.
Lehr- und Lernformen	2V + 1Ü + 1L

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

- Aufbau und Arbeitsweise von diskreten und kontinuierlichen Simulatoren beschreiben
- Simulationsmodelle in verschiedenen Simulationssprachen entwickeln und validieren
- Anwendungsbeispiele und Einsatzmöglichkeiten benennen
- praxisorientierte Kenntnisse in den entsprechenden Simulationsprogrammen vorweisen

Lehrinhalte

- Grundlagen von Simulationssystemen (analytische, stochastische, verteilte und hybride Simulation)
- Simulationssprachen
- Modellbildung und -bewertung
- Simulation als Methode zur Validation und Test
- Tools und Anwendungen, Closed Loop Simulation
- Animation von Simulationsvorgängen

Literatur

- Jazar, R. N. (2017): *Vehicle Dynamics: Theory and Application* (3. Aufl.). Springer.
- Bossel, H. (2014): *Modellbildung und Simulation* (2. Aufl.). Vieweg+Teubner.
- Glöckler, M. (2023): *Simulation mechatronischer Systeme* (3. Aufl.). Springer.
- Kahlert, J. (2004): *Simulation technischer Systeme*. Vieweg.
- Gühmann, C., et al. (2016): *Simulation and Testing for Vehicle Technology*. Springer.

Versionsnummer: 1 (21.02.2024)

2.3.6 Smart IoT (M)

Modulname	Smart IoT (M) / Smart IoT
Modulverantwortliche(r)	T. Dörnbach
Lehrveranstaltung	Smart IoT (M) / Smart IoT
Dauer / Häufigkeit	1 Sem. / Wintersemester
Lehrsprache	Englisch
Prüfungsform/-dauer	PF
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF KT: SCE) • M.Sc. Informatik (PO18) (WPF KT: SCE)
Empf. Voraussetzungen	Grundkenntnisse in Englisch, Grundkenntnisse Internet of Things, Grundkenntnisse Python sollten vorhanden sein.
Lehr- und Lernformen	2SU + 2L

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

- komplexe Architekturen für das Internet der Dinge / Internet of Things (IoT) analysieren, entwerfen, implementieren und weiterentwickeln
- die wichtigsten Technologien, die bei der Realisierung des IoT Anwendung finden, einordnen und auf neue Problemstellungen übertragen
- IoT-Systeme kritisch diskutieren, evaluieren und realisieren

Lehrinhalte

- grundlegende IoT-Technologien und -Architekturen sowie -Geräte
- Flottenmanagement, Data Science und Visualisierung
- Cloud/Fog/Edge Computing
- Smart Home, Smart City
- Blockchain, Generative AI, Federated Learning im IoT-Kontext
- Ethik, Datenschutz und -sicherheit, Nachhaltigkeit
- Vertiefung englischer Sprachkenntnisse

Literatur

Literatur wird in der Lehrveranstaltung bekanntgegeben.

Versionsnummer: 1 (21.02.2024)

2.3.7 Softwareintensive Systeme in der Mobilität (M)

Modulname	Software-intensive Systeme in der Mobilität (M) / Software intensive Systems in Mobility
Modulverantwortliche(r)	C. Führer
Lehrveranstaltung	Software-intensive Systeme in der Mobilität (M) / Software Intensive Systems in Mobility
Dauer / Häufigkeit	1 Sem. / Unregelmäßig
Lehrsprache	Deutsch
Prüfungsform/-dauer	PF(30%)+K1h/M(70%)
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF KT: SCE) • M.Sc. Informatik (PO18) (WPF KT: SCE)
Empf. Voraussetzungen	Sichere Programmierkenntnisse und des Software Engineering zum Beispiel aus einem Bachelorstudium Informatik sollten vorhanden sein.
Lehr- und Lernformen	2V + 1Ü + 1L

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

- besondere Herausforderungen, Lösungsansätze und Standards in der Anwendungsdomäne Mobilität benennen
- aktuelle Anwendungen und zugrundeliegende Architekturen verteilter Systeme in der Mobilität einordnen, Herausforderungen und Lösungsansätze benennen und diskutieren
- komplexe eingebettete und verteilte Softwaresysteme entwerfen und implementieren
- Anforderungen und Architekturansätze analysieren, vergleichen und bewerten

Lehrinhalte

- Typische Anforderungen und Architekturen mit Schwerpunkt auf Infrastruktur- und Onboard-Komponenten von Softwaresystemen in der Mobilität
- Karten und Infrastruktur-Beschreibung, insbesondere topologische Karten
- Herausforderungen und Standards zu Middlewares und Kommunikationsprotokollen
- Besondere Randbedingungen im Entwicklungsprozess
- Anwendungsbeispiele z. B. aus den Bereichen Bahnautomatisierung, Automotive und Avionik

Literatur

Literatur wird in der Lehrveranstaltung bekanntgegeben.

2.3.8 Human-Robot Interaction (M)

Modulname	Human-Robot Interaction (M) / Human-Robot Interaction
Modulverantwortliche(r)	T. Dörnbach
Lehrveranstaltung	Human-Robot Interaction (M) / Human-Robot Interaction
Dauer / Häufigkeit	1 Sem. / Sommersemester
Lehrsprache	Englisch
Prüfungsform/-dauer	PF
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF KT: SCE) • M.Sc. Informatik (PO18) (WPF KT: SCE)
Empf. Voraussetzungen	Grundkenntnisse in Englisch sind erforderlich, Grundkenntnisse Robotik oder autonome Systeme sollten vorhanden sein.
Lehr- und Lernformen	2SU + 2L

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

- die Prinzipien unterschiedlicher Interaktionsformen einordnen, analysieren und auf neue Problemstellungen übertragen
- die wichtigsten technologischen Methoden zur Realisierung einer Mensch-Roboter-Interaktion beschreiben, implementieren und evaluieren
- Forschungsaufgaben, auch interdisziplinär, im Bereich der Mensch-Roboter-Interaktion wissenschaftlich fundiert planen, umsetzen und kritisch evaluieren

Lehrinhalte

- Übersicht über interagierende Systeme und Roboter
- Design von Mensch-Roboter-Interaktion
- Forschungsmethoden in der Mensch-Roboter-Interaktion
- Räumliche Interaktion
- Nonverbale Interaktion
- Verbale Interaktion
- Emotionen
- interaktive Roboter-Lernmethoden
- Vertiefung englischer Sprachkenntnisse

Literatur

- Krug, S. (2014): *Don't Make Me Think, Revisited: a Common Sense Approach to Web Usability* (3. Aufl.). New Riders.
- Hoffmann, G., Zhao, X. (2020): *A Primer for Conducting Experiments in Human-Robot Interaction*. null.

2.4 Grundlagenteil überfachliche Kompetenzen

2.4.1 Entwicklung digitaler Geschäftsmodelle (M)

Modulname	Entwicklung digitaler Geschäftsmodelle (M) / Development of Digital Business Models
Modulverantwortliche(r)	W. Ludwig
Lehrveranstaltung	Entwicklung digitaler Geschäftsmodelle (M) / Development of Digital Business Models
Dauer / Häufigkeit	1 Sem. / Unregelmäßig
Lehrsprache	Deutsch
Prüfungsform/-dauer	PF
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 60% Selbststudium
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPÜ) • M.Sc. Informatik (PO18) (WPF)
Empf. Voraussetzungen	-
Lehr- und Lernformen	4SU

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

- Grundbegriffe der Geschäftsmodellierung benennen
- Komponenten von Geschäftsmodellen beschreiben
- Methoden zur Entwicklung von Geschäftsmodellen anwenden
- Komponenten und Erfolgsfaktoren bestehender digitaler Geschäftsmodelle analysieren und bewerten
- neue digitale Geschäftsmodelle für vorgegebene oder selbst gewählte Gegenstandsbereiche entwickeln
- besser kooperativ zusammenarbeiten
- besser auf fachlicher Ebene kommunizieren

Lehrinhalte

- Begriff des Geschäftsmodells
- Komponenten von Geschäftsmodellen
- Ansätze und Methoden zur Entwicklung von Geschäftsmodellen
- Digitale Geschäftsmodelle
 - Merkmale digitaler Geschäftsmodelle
 - Komponenten und Erfolgsfaktoren digitaler Geschäftsmodelle
 - Ansätze und Methoden zur Entwicklung digitaler Geschäftsmodelle
- Analyse und Bewertung ausgewählter Fallstudien
- Übungen zur Entwicklung digitaler Geschäftsmodelle

Literatur

- Gassmann, O., Sutter, P. (2023): *Digitale Transformation gestalten. Geschäftsmodelle, Erfolgsfaktoren, Checklisten* (3. Aufl.). Hanser.
- Meinhardt, S., Pflaum, A. (2019): *Digitale Geschäftsmodelle*. Springer.
- Osterwalder, A., Pigneur, Y. (2011): *Business Model Generation*. Campus.

2.4.2 Gesprächs- und Verhandlungsführung - Leitung von Arbeitsgruppen (M)

Modulname	Gesprächs- und Verhandlungsführung - Leitung von Arbeitsgruppen (M) / Conduction of Conversations and Negotiations
Modulverantwortliche(r)	Studiendekan
Lehrveranstaltung	Gesprächs- und Verhandlungsführung - Leitung von Arbeitsgruppen (M) /
Dauer / Häufigkeit	1 Sem. / Unregelmäßig
Lehrsprache	Deutsch
Prüfungsform/-dauer	H
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPÜ) • M.Sc. Informatik (PO18) (WPF)
Empf. Voraussetzungen	-
Lehr- und Lernformen	4SU

Qualifikationsziele

Das Seminar vermittelt schrittweise Leitlinien für das Organisieren, Leiten und Führen geschäftlicher Gespräche.

Lehrinhalte

- den eigenen Kommunikationsstil kennen
- verbale und nonverbale Gesprächstechniken kennenlernen, bewusst, strukturiert und gezielt einsetzen
- Besprechungen planen, durchführen und nachbearbeiten
- Argumentationsstrategien
- Konfliktgespräche führen
- Verhandeln nach dem Harvard-Konzept
- typische Verhandlungsfehler erkennen und vermeiden

Literatur

Literatur wird in der Lehrveranstaltung bekanntgegeben.

2.5 Grundlagenteil ohne Schwerpunktzuordnung

2.5.1 Programmierparadigmen C++ (M)

Modulname	Programmierparadigmen C++ (M) / Programming Paradigms in C++
Modulverantwortliche(r)	Studiendekan
Lehrveranstaltung	Programmierparadigmen C++ (M) / Programming Paradigms in C++
Dauer / Häufigkeit	1 Sem. / Unregelmäßig
Lehrsprache	Deutsch
Prüfungsform/-dauer	
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 40% Kontakt- und 50% Selbststudium, 10% Prüfungsvorbereitung
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (WPF) • M.Sc. Informatik (PO18) (WPF)
Empf. Voraussetzungen	Programmieren in C sowie in einer objektorientierten Programmiersprache wie Java sollte beherrscht werden.
Lehr- und Lernformen	4SU

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

- die verschiedenen Programmierparadigmen und C++ verstehen und selbstständig anwenden
- C++ Software mit Test Driven Development entwickeln

Lehrinhalte

- Einführung, Worthäufigkeiten in einer Datei zählen: Datentypen, Steueranweisungen
- Funktionen und Strukturen
- Include-Wächter, Header-Dateien, Werte- und Referenz-Semantik, Zeiger, Heap- und Stackspeicher
- Klassen als abstrakte Datentypen
- tiefe und flache Kopie: Kopierkonstruktor, Nutzung von LogTrace
- Operatoren, Verschiebeoperatoren
- Templates
- Programmieren mit der STL (Standard Template Library)
- Polymorphie, Intelligente Zeiger
- Container
- Lambda-Ausdrücke

Literatur

Literatur wird in der Lehrveranstaltung bekanntgegeben.

2.6 Qualifikationsmodule

2.6.1 Masterseminar (M)

Modulname	Masterseminar (M) / Master Seminar
Modulverantwortliche(r)	Studiendekan
Lehrveranstaltung	Masterseminar (M) / Master seminar
Dauer / Häufigkeit	1 Sem. / Jedes Semester
Lehrsprache	Deutsch
Prüfungsform/-dauer	R
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 10% Kontakt- und 90% Selbststudium
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (Pflicht) • M.Sc. Informatik (PO18) (Pflicht)
Empf. Voraussetzungen	-
Lehr- und Lernformen	S

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

- selbstständig über ein fortgeschrittenes auszuwählendes Thema der Informatik referieren
- aktuelle, fortgeschrittene Themen der Informatik diskutieren
- Inhalte verständlich und rhetorisch/gestalterisch angemessen präsentieren
- das Thema anhand geeigneter erläutern
- Zusammenhänge zu anderen Vortragsthemen oder Themen des Informatikstudiums herstellen
- angemessenes Feedback zu Vorträgen geben, aufgreifen und umsetzen

Lehrinhalte

Aus aktuellen Themen der Informatik wählen alle Studierenden ihr zu referierendes Gebiet und präsentieren ihre inhaltliche Ausarbeitung.

Literatur

Die Literatur wird von der/dem Studierenden mit Unterstützung der/des Dozent/-in selbst zusammengestellt.

2.6.2 Masterprojekt (M)

Modulname	Masterprojekt (M) / Master Project
Modulverantwortliche(r)	Studiendekan
Lehrveranstaltung	Masterprojekt (M) / Master project
Dauer / Häufigkeit	1 Sem. / Jedes Semester
Lehrsprache	Deutsch
Prüfungsform/-dauer	PA
Leistungspunkte	5 ECTS
Aufwand	150h, davon ca. 10% Kontakt- und 90% Selbststudium
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (Pflicht, alternativ kann das Große Masterprojekt gewählt werden) • M.Sc. Informatik (PO18) (Pflicht)
Empf. Voraussetzungen	-
Lehr- und Lernformen	Projektarbeit

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

- ein forschungsnahes Projekt selbstständig erarbeiten und über ein fortgeschrittenes, auszuwählendes Thema der Informatik referieren
- gute Präsentationsstile und Systematiken verwenden
- aktuelle, fortgeschrittene Themen der Informatik in Forschungsgruppen diskutieren

Lehrinhalte

Eigenverantwortliches und wissenschaftlich fundiertes Bearbeiten und Dokumentieren eines Projektes der Informatik, dabei eigenverantwortliche Vertiefung bestehenden theoretischen Wissens.

Literatur

Die Literatur wird von der/dem Studierenden mit Unterstützung der/des Dozent/-in selbst zusammengestellt.

2.6.3 Großes Masterprojekt (M)

Modulname	Großes Masterprojekt (M) / Extensive Master Project (M)
Modulverantwortliche(r)	Studiendekan
Lehrveranstaltung	Großes Masterprojekt (M) / Extensive Master Project (M)
Dauer / Häufigkeit	1 Sem. / Jedes Semester
Lehrsprache	Deutsch
Prüfungsform/-dauer	PA
Leistungspunkte	15 ECTS
Aufwand	450h, davon ca. 10% Kontakt- und 90% Selbststudium
Verwendbarkeit	<ul style="list-style-type: none"> • M.Sc. Informatik (PO25) (Alternative zum Masterprojekt)
Empf. Voraussetzungen	-
Lehr- und Lernformen	Projektarbeit

Qualifikationsziele

Studierende

- referieren zu und erarbeiten ein forschungsnahes Projekt, selbstständig, über ein fortgeschrittenes auszuwählendes Thema der Informatik
- verwenden gute Präsentationsstile und Systematiken
- diskutieren aktuelle, fortgeschrittene Themen der Informatik in Forschungsgruppen

Lehrinhalte

Eigenverantwortliches und wissenschaftlich fundiertes Bearbeiten und Dokumentieren eines umfangreichen Projektes der Informatik, dabei eigenverantwortliche Vertiefung bestehenden theoretischen Wissens.

Literatur

Die Literatur wird von der/dem Studierenden mit Unterstützung der/des Dozent/-in selbst zusammengestellt.

Versionsnummer: 1 (22.02.2024)

2.7 Masterarbeit mit Kolloquium

2.7.1 Masterarbeit mit Kolloquium (M)

Modulname	Masterarbeit mit Kolloquium (M) / Master Thesis with Colloquium
Modulverantwortliche(r)	Studiendekan
Lehrveranstaltung	Masterarbeit mit Kolloquium (M) / Master Thesis with Colloquium
Dauer / Häufigkeit	1 Sem. / Jedes Semester
Lehrsprache	Deutsch
Prüfungsform/-dauer	
Leistungspunkte	27 ECTS (Masterarbeit) + 3 ECTS (Kolloquium)
Aufwand	900h
Verwendbarkeit	• M.Sc. Informatik (PO25) (Pflicht)
Empf. Voraussetzungen	-
Lehr- und Lernformen	Wissenschaftliche Tätigkeit, Projektarbeit

Qualifikationsziele

Nach erfolgreichem Absolvieren des Moduls können Studierende:

- ein komplexes praxisbezogenes oder ein forschungsorientiertes Problem aus dem Fachgebiet der Informatik innerhalb einer vorgegebenen Frist auf der Grundlage wissenschaftlicher Methoden selbständig bearbeiten, in einer schriftlichen wissenschaftlichen Ausarbeitung dokumentieren und die Arbeitsergebnisse in einem Fachgespräch präsentieren und verteidigen
- das erworbene Fachwissen in einem forschungs- oder praxisbezogenen Umfeld anwenden und selbständig um das für die Bearbeitung des Problems notwendige Anwendungs- und Spezialwissen ergänzen und vertiefen

Lehrinhalte

Eigenverantwortliches Bearbeiten einer wissenschaftlichen Themenstellung der Informatik (gegebenenfalls auf Basis eines entsprechenden Projekts), dabei eigenverantwortliche Vertiefung bestehenden theoretischen Wissens.

Literatur

Die Literatur wird von der/dem Studierenden selbst zusammengestellt.

3 Modulübersichtstabelle

Modul / Lehrveranstaltung	Sem.	ECTS	Prüfungsform	Lehr- und Lernform
Grundlagenteil / Kompetenzteil Software Engineering				
Architekturen moderner Informationssysteme (M)	1-3	5	K1,5h	4SU
Cloud Native Computing (M)	1-3	5	EA(30%) +K1,5h(70%)	2V + 1Ü + 1L
Effiziente Algorithmen (M)	1-3	5	K1,5h/M/PA	4SU
Entwicklung großer Anwendungssysteme (M)	1-3	5	K1,5h	3V + 1Ü
Formale Methoden (M)	1-3	5	K1,5h(50%) +PA(50%)	2V + 1Ü + 1L
IT-Management (M)	1-3	5	K1,5h/PA	2V + 1Ü + 1L
Modellgetriebene Softwareentwicklung (M)	1-3	5	EA(50%) +K1,5h(50%)	2V + 1Ü + 1L
User Interfaces für Mobile Systeme (M)	1-3	5	(R(15%)+PA(35%) +K1,5h(50%)) /(K1,5h(50%) +PA(50%))	4SU
Werkzeuge der Java-Plattform (M)	1-3	5	K1,5h	3V + 1Ü
Softwareintensive Systeme in der Mobilität (M)	1-3	5	PF(30%)+K1h /M(70%)	2V + 1Ü + 1L
Grundlagenteil / Kompetenzteil Data Science				
Statistische Methoden (M)	1-3	5		2V + 1Ü + 1L
Heuristische Suche (M)	1-3	5	EA(40%) +K1,5h(60%)	2V + 2L
Neuronale Netze und Deep Learning (M)	1-3	5	K1,5h	2V + 1L + 1Ü
Automatische Sprachverarbeitung (M)	1-3	5	K1,5h	2V + 1L + 1Ü

Datenbanktechnologien (M)	1-3	5	EA(10%) +K1,5h(90%)	3SU + 1Ü
Big Data (M)	1-3	5	EA(40%) +K1,5h(60%)	3SU + 1Ü
Angewandte Kryptographie (M)	1-3	5	PF(20%) +K1,5h(80%)	2V + 1Ü + 1P
Machine Learning (M)	1-3	5	K1,5h	4SU
Stream Processing (M)	1-3	5	K1,5h	4SU
Grundlagenteil / Kompetenzteil Systems & Computer Engineering				
Automotive Systems (M)	1-3	5	EA(30%)+K1h(70%)	2V + 1Ü + 1L
Autonomous Systems (M)	1-3	5	K1h(50%)+PF(50%)	FC, 2SU + 2L
Robotik (M)	1-3	5	K1h(50%)+PF(50%)	FC, 2SU + 2L
Robuste Systeme (M)	1-3	5	PF(30%)+K1h /M(70%)	2V + 1Ü + 1L
Simulation und Verifikation (M)	1-3	5	EA(30%)+K1h(70%)	2V + 1Ü + 1L
Smart IoT (M)	1-3	5	PF	2SU + 2L
Softwareintensive Systeme in der Mobilität (M)	1-3	5	PF(30%)+K1h /M(70%)	2V + 1Ü + 1L
Human-Robot Interaction (M)	1-3	5	PF	2SU + 2L
Grundlagenteil überfachliche Kompetenzen				
Entwicklung digitaler Geschäftsmodelle (M)	1-3	5	PF	4SU
Gesprächs- und Verhandlungsführung - Leitung von Arbeitsgruppen (M)	1-3	5	H	4SU
Grundlagenteil ohne Schwerpunktuordnung				
Programmierparadigmen C++ (M)	1-3	5		4SU
Qualifikationsmodule				
Masterseminar (M)	3	5	R	S
Masterprojekt (M)	3	5	PA	Projektarbeit

Großes Masterprojekt (M)	3	15	PA	Projektarbeit
Masterarbeit mit Kolloquium				
Masterarbeit mit Kolloquium (M)	4	27+3		Wissenschaftliche Tätigkeit, Projektarbeit

4 Änderungshistorie