

Modulhandbuch für den Bachelor-Studiengang Informatik

Für die Prüfungsordnung 2021

Institut für Informatik
der Mathematisch-Naturwissenschaftliche Fakultät
der Heinrich-Heine-Universität

Herausgegeben vom
Prüfungsausschuss des Bachelor-Studiengangs Informatik

Aktualisiert am 16.02.2024

Vorwort

Das Modulhandbuch soll eine Orientierung über den grundständigen Bachelor-Studiengang Informatik liefern. Insbesondere soll es die Wahl der Lehrveranstaltungen erleichtern und die Organisation des Studiums unterstützen.

Das Modulhandbuch führt die gängigen Lehrveranstaltungen auf. Es ist jedoch keine vollständige, abschließende oder endgültige Auflistung. Vielmehr wird das Modulhandbuch kontinuierlich aktualisiert und gibt so die Entwicklung in Forschung und Lehre am Institut für Informatik wieder.

Im Abschnitt Teilnahmevoraussetzungen unterscheiden wir zwischen formalen und inhaltlichen Voraussetzungen. Wenn Sie formale Voraussetzungen nicht erfüllen, dürfen Sie sich zu diesem Modul nicht anmelden. Module mit inhaltlichen Voraussetzungen, setzen bestimmte Kenntnisse voraus. Wenn Sie diese Kenntnisse nicht mitbringen, müssen Sie diese selbstständig erwerben. In der Regel empfehlen wir dazu, die entsprechenden Module zu absolvieren.

Beachten Sie jedoch, dass in allen Fragen zum Studium sowie zum Prüfungswesen die Bachelor-Prüfungsordnung des Fachs Informatik maßgeblich ist.

Düsseldorf, 16.02.2024

Der Prüfungsausschuss für den Bachelor-Studiengang Informatik

Inhaltsverzeichnis

| | |
|---|-----------|
| Vorwort | 2 |
| Qualifikationsziele des Studiengangs Bachelor Informatik | 5 |
| Wissenschaftliche Befähigung und Befähigung zu einer qualifizierten Erwerbstätigkeit | 5 |
| Persönlichkeitsentwicklung | 5 |
| Studienverlaufspläne | 6 |
| Pflichtmodule der Informatik (1.–4. Fachsemester) (en. Compulsory Modules in Computer Science) | 7 |
| Algorithmen und Datenstrukturen (en. Algorithms and Data Structures) | 8 |
| C-Programmierung für Algorithmen und Datenstrukturen (en. C Programming for Algorithms and Data Structures) | 10 |
| Data Science (en. Data Science) | 12 |
| Datenbanken: Eine Einführung (en. Databases: An Introduction) | 14 |
| Grundlagen der Computernetzwerke (en. Foundations of Computer Networks) | 16 |
| Programmierpraktikum 1 (en. Programming Project 1) | 18 |
| Programmierpraktikum 2 (en. Programming Project 2) | 20 |
| Programmierung (en. Programming) | 22 |
| Rechnerarchitektur (en. Computer Architecture) | 24 |
| Theoretische Informatik (en. Theoretical Computer Science) | 26 |
| Wissenschaftliches Arbeiten (en. Scientific Methods) | 28 |
| Pflichtmodule der Mathematik (en. Compulsory Modules in Mathematics) | 30 |
| Mathematik für Informatik 1 (en. Mathematics for Computer Science 1) | 31 |
| Mathematik für Informatik 2 (en. Mathematics for Computer Science 2) | 33 |
| Mathematik für Informatik 3 (en. Mathematics for Computer Science 3) | 35 |
| Lehreinheiten für den Wahlbereich (en. Courses for elective areas) | 37 |
| Algorithmen in der Bioinformatik (en. Algorithms in Bioinformatics) | 38 |
| Algorithmische Geometrie (en. Computational Geometry) | 40 |
| Algorithmische Komplexitätstheorie (en. Computational Complexity Theory) | 42 |
| Angewandte Algorithmik (en. Applied Algorithmics) | 44 |
| Betriebssysteme und Systemprogrammierung (en. Operating Systems and System Programming) .. | 46 |
| Competitive Programming A (en. Competitive Programming A) | 48 |
| Competitive Programming B (en. Competitive Programming B) | 50 |
| Compilerbau (en. Compiler Construction) | 52 |
| Computer-gestützte Beweisführung (en. Computer-aided reasoning) | 54 |
| Data Science 2 (en. Data Science 2) | 56 |
| Datenbanken: Weiterführende Konzepte (en. Databases: Further Concepts) | 58 |
| Datenvisualisierung (en. Data Visualization) | 60 |
| Datenvisualisierung (en. Data Visualization) | 61 |
| Digitale Innovation: Von der Idee zur Wirkung (en. Digital Innovation: From Idea to Impact) .. | 62 |
| Einführung in algorithmische Spieltheorie (en. Introduction to Algorithmic Game Theory) .. | 64 |
| Einführung in Deep Learning (en. Introduction to Deep Learning) | 66 |
| Einführung in die Funktionale Programmierung (en. Introduction to Functional Programming) .. | 68 |
| Einführung in die naturwissenschaftliche Informatik (en. Introduction to Scientific Computer Science) . | 70 |
| Graphenalgorithmen 1 (en. Algorithms for Graphs 1) | 72 |
| Grundlagen Verteilter Systeme (en. Foundations of Distributed Systems) | 74 |

| | |
|--|------------|
| Introduction to Logic Programming (en. Introduction to Logic Programming) | 76 |
| Introduction to Modelling metabolic networks (en. Introduction to Modelling metabolic networks) | 78 |
| Kombinatorische Algorithmen für Clusteringprobleme (en. Combinatorial Algorithms for Clustering Problems) | 80 |
| Kryptokomplexität 1 (en. Cryptocomplexity 1) | 82 |
| Muster in der Natur: theoretische Hintergründe und Algorithmen (en. Patterns in nature: theoretical background and algorithms) | 84 |
| Präferenzaggregation durch Wählen: Algorithmik und Komplexität (en. Preference Aggregation by Voting: Algorithmics and Complexity) | 86 |
| Randomisierte Algorithmen und Analysetechniken (en. Randomized Algorithms und Analysis) | 88 |
| Statistische Datenanalyse (en. Statistical Data Analysis) | 90 |
| Von der Schaltung zur Software (en. From Circuits to Software) | 92 |
| Seminar (en. Seminar) | 94 |
| Bachelor-Seminar (en. Bachelor's-Seminar) | 95 |
| Bachelorarbeit (en. Bachelor Thesis) | 97 |
| Abschlussarbeit (en. Thesis) | 98 |
| Abschlussseminar (en. Final Seminar) | 99 |
| Nicht mehr angebotene Module | 100 |
| Bachelor-Seminar: Einführung in die Blockchaintechnologie (en. Bachelor's Seminar: Introduction to blockchain technology) | 101 |
| Bachelor-Seminar: Programmiersprachen (en. Bachelor's-Seminar: Programming Languages) | 103 |
| Bachelor-Seminar: Überblick Künstliche Intelligenz (en. Bachelor's-Seminar: Introduction to Artificial Intelligence) | 105 |
| Kollektive Entscheidungen (en. Collective Decisions) | 107 |
| Matching (en. Matching) | 109 |

Qualifikationsziele des Studiengangs Bachelor Informatik

Nach Abschluss des Studiums sollen die Studierenden folgende Qualifikationsziele erreicht haben.

Wissenschaftliche Befähigung und Befähigung zu einer qualifizierten Erwerbstätigkeit

Absolvent*innen

- können erlernte Methoden und Verfahren der Informatik (z. B. systematische Problemdefinitionen, algorithmische Problemlösungen, logische Beweisverfahren, Evaluationsverfahren) vergleichen und anwenden.
- sind in der Lage, Anwendungen mit neuen Software-Technologien zu realisieren. Dies umfasst die Konzeption und die Implementierung sowie die Entwicklung von Testverfahren.
- sind mit einem logisch-analytischen, systemischen Denkansatz vertraut, der es ihnen ermöglicht, neuartige Problemstellungen, beispielsweise im Bereich Data Science und Machine Learning, zu untersuchen und Lösungen abzuleiten.
- können ihr erlerntes Wissen sowie Forschungsmethoden unter Berücksichtigung allgemein anerkannter Grundsätze guter wissenschaftlicher Praxis anwenden, um Forschungsprobleme zu lösen und den Nutzen der Ergebnisse zu beurteilen.

Persönlichkeitsentwicklung

Absolvent*innen

- können ihre eigenen Fähigkeiten im Hinblick auf Aspekte wie (Daten-) Analyse, Software-Entwicklung oder Kommunikation einschätzen und haben bereits Ideen zu deren Weiterentwicklung. Sie können sich selbständig neues Fachwissen erschließen.
- können Verantwortung für sich selbst bzw. die eigene Aufgabe in einer Gruppe übernehmen.
- haben ein berufliches Selbstbild (z. B. als Datenwissenschaftler*innen oder Softwarearchitekt*innen) entwickelt und kennen alternative Entwürfe.
- können informatische Probleme und passende Lösungen fachgerecht kommunizieren und in Gruppen diskutieren, wie sie es im Rahmen von Übungen und Seminaren erlernt haben.
- kennen die Herangehensweisen an ethische Fragen und Herausforderungen aus Sicht der Informatik sowie die gesellschaftliche, kulturelle und politische Bedeutung ihrer Fachdisziplin.

Studienverlaufspläne

Auf den Webseiten des Instituts für Informatik finden Sie [Studienverlaufspläne](#). Diese dienen der Orientierung und helfen Ihnen bei der Planung Ihres Studiums. Es gibt verschiedene Varianten der Studienverlaufspläne:

- Vollzeit, mit Mathematik für Informatik
- Vollzeit, mit Mathematik der Mathematik
- Teilzeit, mit Mathematik für Informatik

Sollten Sie Hilfe bei Ihrer Studienplanung benötigen, wenden Sie sich bitte an die Studienberatung unter studieninteressierte@cs.uni-duesseldorf.de.

Pflichtmodule der Informatik (1.–4. Fachsemester) (en. Compulsory Modules in Computer Science)

Algorithmen und Datenstrukturen (en. Algorithms and Data Structures)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------------|------------------------------------|--------------------|
| 10 LP | 300 Stunden | 90 Stunden | 210 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (4 SWS) Übung (2 SWS) | jedes Wintersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Das Modul befasst sich mit einer Auswahl von grundlegenden Algorithmen und Datenstrukturen aus Theorie und Praxis. Anhand der vorgestellten Algorithmen und Datenstrukturen wird erläutert, wie der Ressourcenverbrauch (Rechenzeit- und Speicherplatzbedarf) theoretisch analysiert und vorhergesagt werden kann.

- Algorithmen und ihre formalen Grundlagen
- Rechenmodelle, Effizienzmaße
- Suchstrategien (Binärsuche)
- Sortierverfahren (Quicksort, Heapsort, Mergesort, ...)
- Grundlegende Datenstrukturen (Arraylisten, verkettete Listen, Stacks und Queues)
- Suchbäume (Binärbäume, Balancierte Suchbäume)
- Dictionaries (offene Hashverfahren, dynamische Hashverfahren)
- Prioritätswarteschlangen (Binäre Heaps)
- Verwaltung von Mengensystemen (Union Find)
- Amortisierte Laufzeitanalyse
- Graphenalgorithmen (Tiefensuche, Breitensuche, Spannbäume, Kürzeste Wege)
- Entwurfsmuster (Greedyalgorithmen, Divide-and-Conquer, Dynamische Programmierung)
- Grenzen effizienter Algorithmen (Ausblick)

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- die vorgestellten Algorithmen/Datenstrukturen anwenden, analysieren und ihre Funktionsweise erklären,
- einen geeigneten Algorithmus/eine geeignete Datenstruktur für eine Problemstellung erkennen und aus einem Repertoire auswählen,
- aus einer natürlichsprachlichen Beschreibung eine Spezifikationen eines Algorithmus oder einer Datenstrukturen entwickeln und bestehende Spezifikationen erklären und Fragen dazu beantworten
- den Ressourcenbedarf von Algorithmen und Datenstrukturen mit den fundamentalen Techniken der Vorlesung analysieren, vorhersagen und vergleichen,
- nachweisen, dass ein Algorithmus/eine Datenstruktur korrekt arbeitet und andere davon überzeugen und
- die vorgestellten Algorithmen und Datenstrukturen anpassen und kombinieren.

Literatur

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein: Algorithmen – Eine Einführung. De Gruyter Oldenbourg. 2017. 4. Auflage.
- Robert Sedgewick, Kevin Wayne: Algorithmen. Pearson Studium. 2014. 4. Auflage.
- Thomas Ottmann, Peter Widmayer: Algorithmen und Datenstrukturen. Spektrum. 2012. 5. Auflage.

- Jon Kleinberg, Eva Tardos: Algorithm Design. Addison Wesley. 2006.

Verwendbarkeit des Moduls

- Pflichtbereich Bachelor-Studiengang PO 2021
- Pflichtbereich Bachelor-Studiengang PO 2016, 2013
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik
- Modul CL6 im Bachelor-Studiengang Computerlinguistik

Teilnahmevoraussetzungen

- Inhalte des Moduls *Mathematik für Informatik 1*

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive Mitarbeit in den Übungen
- Abgabe der Hausaufgaben
- schriftliche Klausur (i. d. R. 90 Minuten) oder mündliche Prüfung am Ende des Semesters

Modulbeauftragte und hauptamtlich Lehrende

Dr. Daniel Schmidt, Prof. Dr. Gunnar W. Klau, Prof. Dr. Melanie Schmidt

C-Programmierung für Algorithmen und Datenstrukturen (en. C Programming for Algorithms and Data Structures)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) | jedes Sommersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Dieses Modul baut auf dem Modul *Algorithmen und Datenstrukturen* auf und ist der Implementierung der dort vermittelten Konzepte in der Programmiersprache C gewidmet. Konkret werden folgende Themen behandelt:

- Einführung in C
- Grundlegende Datenstrukturen (Stacks, doppelt verkettete Listen, Bäume)
- Suchbäume
- Heaps
- Quicksort
- Hashing
- Dynamische Graphdatenstruktur und Dijkstras Algorithmus

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- zentrale Algorithmen und Datenstrukturen in der Programmiersprache C unter Berücksichtigung dynamischer Speicherverwaltung implementieren und
- Werkzeuge für typische Aufgaben bei der C-Programmierung (Speicherverwaltung, Debugging, Build-Prozesse) verwenden.

Literatur

- R. Sedgewick: *Algorithmen in C*. Pearson. 2005.
- R. Seacord: *Effective C*. No Starch Press. 2020.
- G. Gonzales-Morris und I. Horton: *Beginning C: From Beginner to Pro*. Apress. 2020.

Verwendbarkeit des Moduls

- Pflichtbereich Bachelor-Studiengang PO 2021
- Hardwarenahe Programmierung PO 2013, 2016

Teilnahmevoraussetzungen

- Inhaltlich: Inhalte der Module *Algorithmen und Datenstrukturen* und *Programmierung*

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive und erfolgreiche Mitarbeit in den Übungen
- schriftliche Prüfung (Klausur, i. d. R. 90 Minuten)

Modulbeauftragte und hauptamtlich Lehrende

Dr. Daniel Schmidt, Prof. Dr. Gunnar W. Klau, Prof. Dr. Melanie Schmidt

Data Science (en. Data Science)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------------|------------------------------------|--------------------|
| 10 LP | 300 Stunden | 90 Stunden | 210 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (4 SWS) Übung (2 SWS) | jedes Sommersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Data Science ist die Anwendung statistischer Methoden und Methoden des maschinellen Lernens auf Daten jedweder Art mit Hilfe eines Rechners zur Modellierung von Systemen und Vorhersage von Verhalten.

- Wahrscheinlichkeitstheorie (Diskrete und kontinuierliche Verteilungen, Bayes Theorem, Unabhängigkeit, Normalverteilung, Multivariate Verteilungen, Transformationen)
- Maschinelles Lernen (Daten und Modelle, Schätztheorie, Klassifikation, Regression, Dimensionalitätsreduktion, Clusteranalyse)
- Python-Pakete für Data Science (Numpy, Matplotlib, Pandas, Scikit-learn)

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- einen Datensatz explorativ analysieren mit Hilfe der Programmiersprache Python,
- Daten zur Weiterverarbeitung transformieren,
- Forschungsfragen an einen Datensatz stellen,
- Teildatensätze zur Weiterverarbeitung wählen,
- geeignete Datensätze zusammenführen,
- Erkenntnisse in Daten mit statistischen Fachbegriffen beschreiben und visualisieren mit Hilfe der Pakete Pandas und Matplotlib,
- Datenschutz-Problematiken bei der Datenverarbeitung berücksichtigen,
- ethische Fragen bei der Datenverarbeitung berücksichtigen,
- statistische Modelle anhand von Daten aufstellen mit Hilfe des Pakets Scikit-Learn,
- die Güte von Modellen bewerten und
- statistische Modelle zur Klassifikation und Vorhersage weiterer Daten nutzen.

Literatur

- Georgii: Stochastik: Einführung In Die Wahrscheinlichkeitstheorie Und Statistik. De Gruyter. Berlin, 2015. 5. Auflage
- VanderPlas: Python Data Science Handbook. O'Reilly Media, Inc. Sebastopol, 2016. 1st edition
- Grus: Data Science from Scratch: First Principles with Python. O'Reilly UK Ltd. UK, 2019. 2nd edition
- Deisenroth et. al.: Mathematics for Machine Learning. CUP. Cambridge, 2020. 1st edition

Verwendbarkeit des Moduls

- Pflichtbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

- Inhaltlich: Inhalte der Module *Mathematik für Informatik 1–3*

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive und erfolgreiche Mitwirkung in den theoretischen und praktischen Übungen
- Bestehen der schriftlichen Prüfung (Klausur, i. d. R. 90 Minuten)

Modulbeauftragte und hauptamtlich Lehrende

Dr. Konrad Völkel, Prof. Dr. Milica Gasic

Datenbanken: Eine Einführung (en. Databases: An Introduction)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) | jedes Sommersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Dieses Modul vermittelt grundlegende theoretische und praktische Kenntnisse zu Datenbanken. Der Schwerpunkt liegt dabei auf den verschiedenen Phasen des Datenbankentwurfs sowie Anfragen in SQL und den dafür erforderlichen Grundlagen.

- Architektur von Datenbanksystemen; 3-Ebenen-Schemaarchitektur
- Aufgaben eines Datenbanksystems
- Daten(bank)modelle; insb. Relationales Modell
- konzeptioneller und logischer Datenbankentwurf (Phasenmodell, ER-Modell, Transformation in relationale Modell); Normalisierung (funktionale Abhängigkeiten, Normalformen, Synthesealgorithmus)
- Anfragesprachen für relationale Datenbanken und ihre Grundlagen; relationale Algebra, Kalküle, SQL
- Concurrency Control: Probleme des Mehrbenutzerbetriebs, Transaktionen und Schedules, Serialisierbarkeit

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- die Aufgaben und Funktionen eines Datenbanksystems benennen und erläutern,
- selbstständig Datenbanken entwerfen,
- Datenbankanfragen in verschiedenen Formalismen (insb. relationale Algebra und SQL) formulieren und zwischen den Formalismen übersetzen und
- Transaktionsschedules auf Serialisierbarkeit überprüfen.

Literatur

- G. Saake, K.-U. Sattler, A. Heuer: Datenbanken – Konzepte und Sprachen. mitp Verlag. 2018. 6. Auflage
- Kemper, A. Eickler: Datenbanksysteme – Eine Einführung. Oldenbourg Verlag. 2015. 10. Auflage
- G. Vossen: Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme. Oldenbourg Verlag. 2008.
- R. Elmasri, S.B. Navath: Fundamentals of Database Systems. Pearson. 2016. 7th edition
- H. Garcia-Molina, J.D. Ullman, J. Widom: Database Systems: The Complete Book. Pearson. 2009. 2nd edition

Verwendbarkeit des Moduls

- Pflichtbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

- Modul CL6 im Bachelor-Studiengang Computerlinguistik

Teilnahmevoraussetzungen

- keine

Voraussetzungen für die Vergabe von Leistungspunkten

- Aktive und erfolgreiche Teilnahme an den Übungen (in der Regel mit Hausaufgaben)
- schriftliche Prüfung (Klausur, i. d. R. 60 Minuten)

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Stefan Conrad, Dr. Leonie Selbach

Grundlagen der Computernetzwerke (en. Foundations of Computer Networks)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) | jedes Sommersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Das Modul *Grundlagen der Computernetzwerke* richtet sich an Studierende, die verstehen wollen, wie Computernetzwerke aus technischer Sicht funktionieren. Es werden die grundlegenden Fragestellungen des Gebietes untersucht und es wird besprochen, wie diese Fragestellungen im Internet gelöst sind. Ziel des Moduls ist es, sowohl ein solides allgemeines Basiswissen im Bereich der Computernetzwerke als auch praktisch einsetzbare Kenntnisse zu vermitteln.

- Einleitung und Übersicht
- Anwendungsschicht (World Wide Web/HTTP, File Transfer/FTP, E-Mail/SMTP, Domain Name System/DNS, Socketprogrammierung mit UDP und TCP)
- Transportschicht (Adressierung, zuverlässige Datenübertragung/RDT, Überlastkontrolle, UDP, TCP)
- Netzwerkschicht (Virtuelle Leitungen und Datagrammnetzwerke, Funktionsweise und Aufbau von Routern, Adressierung/DHCP und NAT, das Internetprotokoll/IP, ICMP, Link State Routing und Distance Vector Routing, RIP, OSPF, BGP)
- Sicherungsschicht (Rahmenbildung, Fehlererkennung und Fehlerkorrektur, Medienzugriff in Lokalen Netzen, Adressierung/ARP, Ethernet, Hubs, Switches, PPP, IP over ATM und MPLS, Anwendungen)

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden

- die geschichtete Netzwerkarchitektur des Internets erläutern und die wichtigsten Protokolle in den einzelnen Schichten anwenden,
- die Performance einzelner Protokolle in einfachen Netzwerken berechnen,
- die Adressierungsmechanismen in den verschiedenen Netzwerkschichten erklären und ihr Zusammenspiel diskutieren,
- einfache lokale Netzwerkstrukturen analysieren bzw. für gegebene Anwendungsfälle geeignete Netzwerkstrukturen designen und mit geeigneten Protokollen implementieren,
- grundlegende Methoden anwenden, um die Performance von komplexen Netzwerken zu optimieren, und
- verschiedene Arten von Übertragungsfehlern benennen sowie Möglichkeiten aufzeigen, Übertragungsfehler in den verschiedenen Netzwerkschichten zu erkennen und zu beheben.

Literatur

- James F. Kurose and Keith W. Ross: Computer Networking – A Top-Down Approach Featuring the Internet. Pearson, 2020. 8th Edition

Verwendbarkeit des Moduls

- Pflichtbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016

- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

- keine

Voraussetzungen für die Vergabe von Leistungspunkten

- erfolgreiches Bearbeiten der Übungsaufgaben (50% nötig für Klausurzulassung)
- schriftliche Prüfung (Klausur, i. d. R. 90 Minuten)

Modulbeauftragte und hauptamtlich Lehrende

apl. Prof. Dr. Rudolf Fleischer, Prof. Dr. Martin Mauve

Programmierpraktikum 1 (en. Programming Project 1)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|---|----------------------|------------------------------------|--------------------|
| 10 LP | 300 Stunden | 66 Stunden | 234 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) praktische Übung (1 SWS) | jedes Sommersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Ziel der beiden Module des Programmierpraktikums ist es, Studierende dazu zu befähigen, im Team ein größeres, webbasiertes, qualitativ hochwertiges Informationssystem in Java entwickeln zu können. Das *Programmierpraktikum 1* behandelt folgende Themen:

- Werkzeuge in der Softwareentwicklung (IDE, Buildwerkzeuge, Versionsverwaltung)
- Automatische Softwaretests und testgetriebene Entwicklung
- Code-Qualität und Code Smells
- Prinzipien und Praktiken zur Modularisierung größerer Systeme mithilfe von objektorientierter Programmierung (SOLID-Prinzipien, Information Hiding, Kopplung und Kohäsion, Gesetz von Demeter, Polymorphismus, Dependency Injection)
- Grundlagen des Domain-driven Design (Ubiquitous Language und taktisches Design)
- Systematische Fehlersuche
- Fortgeschrittene Programmierkonzepte in Java (u. a. Streams, Records, Optional, DateTime, ...)

Zusätzliche Bemerkungen

- Die Angabe der SWS zur Kontaktzeit ist nicht genau sondern gerundet.
- Die Vorlesung folgt dem Modell des Flipped Classrooms.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- selbständig ein größeres, wartbares Softwaresystem basierend auf einer Problemstellung entwickeln
- die Qualität von bestehendem Code analysieren und begründet erläutern, wie sich die Qualität auf die Wartbarkeit auswirkt
- die Qualität und Wartbarkeit von bestehendem Code verbessern
- testgetriebenen Code entwickeln
- Java-Sprachfeatures idiomatisch verwenden
- systematisch Fehlerursachen in Code finden und beheben
- Softwareentwicklungswerkzeuge (IDE, Versionsverwaltung, Buildsystem) verwenden

Literatur

- Eigenes Skript

Verwendbarkeit des Moduls

- Pflichtbereich Bachelor-Studiengang PO 2021
- Pflichtbereich Bachelor-Studiengang PO 2016, 2013

- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik
- Modul CL6 im Bachelor-Studiengang Computerlinguistik

Teilnahmevoraussetzungen

- Formal: Erfolgreicher Abschluss des Moduls *Programmierung*

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive und erfolgreiche Mitwirkung in den theoretischen und praktischen Übungen
- schriftliche Prüfung (Klausur, i. d. R. 60 Minuten)

Modulbeauftragte und hauptamtlich Lehrende

Dr. Jens Bendisposto, Dr. Markus Brenneis

Programmierpraktikum 2 (en. Programming Project 2)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|---|----------------------|------------------------------------|--------------------|
| 10 LP | 300 Stunden | 124 Stunden | 176 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) praktische Übung (1 SWS) Blockpraktikum (4 SWS) | jedes Wintersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Ziel der beiden Module des Programmierpraktikums ist es, Studierende dazu zu befähigen, im Team ein größeres, webbasiertes, qualitativ hochwertiges Informationssystem in Java entwickeln zu können. Das *Programmierpraktikum 2* behandelt folgende Themen:

- Webentwicklung (HTTP, HTML, Accessibility, Authentifizierung und Autorisierung, Security, REST, Servlets, Spring Web)
- Anbindung an Datenbanken (JDBC, Spring Data)
- Softwarearchitektur und Architekturmuster
- Integration- und Architektur-Testing (ArchUnit)
- Vorgehensmodelle
- Architekturdokumentation (arc42, UML)

Zusätzliche Bemerkungen

- Die Angabe der SWS zur Kontaktzeit ist nicht genau sondern gerundet.
- Die Vorlesung folgt dem Modell des Flipped Classrooms.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- Webanwendungen implementieren und absichern
- die Barrierefreiheit von Webanwendungen sicherstellen
- datenbankbasierte Informationssysteme entwickeln
- Architekturmuster benennen und anwenden
- Integrationstests für webbasierte Informationssysteme schreiben
- Tests zur Absicherung von Architekturentscheidungen schreiben
- gängige Vorgehensmodelle beschreiben und anwenden
- die Architektur eines Systems dokumentieren

Literatur

- Eigenes Skript

Verwendbarkeit des Moduls

- Pflichtbereich Bachelor-Studiengang PO 2021
- Pflichtbereich Bachelor-Studiengang PO 2016, 2013

Teilnahmevoraussetzungen

- Formal: Erfolgreicher Abschluss des Moduls *Programmierpraktikum 1*

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive und erfolgreiche Mitwirkung in den theoretischen und praktischen Übungen
- aktive und erfolgreiche Mitwirkung im Praktikum
- schriftliche Prüfung (Klausur, i. d. R. 60 Minuten)

Modulbeauftragte und hauptamtlich Lehrende

Dr. Jens Bendisposto

Programmierung (en. Programming)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|--|----------------------|------------------------------------|--------------------|
| 10 LP | 300 Stunden | 120 Stunden | 180 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (4 SWS) Übung (2 SWS) Tutorium (2 SWS) | jedes Wintersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Dieses Modul vermittelt Grundlagen der imperativen, objektorientierten Programmierung exemplarisch anhand der Programmiersprache Java. Dabei werden einführend Aspekte von Algorithmen und Datenstrukturen behandelt. Es wird keine Programmiererfahrung vorausgesetzt.

- Primitive Datentypen und Variablen
- Kontrollstrukturen
- Arrays
- Standardein- und -ausgabe, Dateien
- Programmstrukturen im Speicher (Heap, Stack)
- Rekursion
- Konzepte der Objektorientierung (Klassen, Objekte, Polymorphie, Interfaces, Vererbung)
- Fehlerbehandlung
- Suchen und Sortieren (lineare Suche, binäre Suche, Insertion Sort, Mergesort)
- Dynamische Datenstrukturen (Listen, Binäre Suchbäume, Hashing)

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- gegebenen Java-Programmcode erklären,
- Beschreibungen iterativer/rekursiver Algorithmen in strukturierten Code übersetzen und eigene, einfache Algorithmen entwerfen,
- Standardein- und -ausgabe sowie Dateien zum Einlesen und Ausgeben von Textdaten verwenden,
- objektorientierte Programme schreiben, die Gegebenheiten der echten Welt abbilden,
- Vorteile von Polymorphismus erklären und anwenden,
- die Vor- und Nachteile verschiedener Datenstrukturen (Arrays, einfach verkettete Listen, binäre Suchbäume, Hashsets) erläutern,
- Algorithmen für verschiedene Datenstrukturen (Arrays, einfach verkettete Listen, binäre Suchbäume, Hashsets) entwerfen und implementieren, und
- Compilzeitfehler beheben und Laufzeitfehler angemessen behandeln.

Literatur

- R. Schiedermeier: Programmieren mit Java. Pearson Studium. München, 2010. 2. aktualisierte Ausgabe
- C. Ullenboom: Java ist auch eine Insel. Rheinwerk Computing. Bonn, 2021. 16. Auflage
- R. Sedgewick & K. Wayne: Introduction to Programming in Java. Addison-Wesley, United States, 2007. 1st Edition

Verwendbarkeit des Moduls

- Pflichtbereich Bachelor-Studiengang PO 2021
- Pflichtbereich Bachelor-Studiengang PO 2016, 2013
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik
- Modul I: Informatik im Bachelor-Studiengang Computerlinguistik

Teilnahmevoraussetzungen

- keine

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive und erfolgreiche Mitwirkung in den theoretischen und praktischen Übungen
- schriftliche Prüfung (Klausur, i. d. R. 120 Minuten)

Modulbeauftragte und hauptamtlich Lehrende

Dr. Markus Brenneis, Prof. Dr. Michael Schöttner

Rechnerarchitektur (en. Computer Architecture)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|--|----------------------|------------------------------------|--------------------|
| 7 LP | 210 Stunden | 90 Stunden | 120 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (3 SWS) Übung (2 SWS) Tutorium (1 SWS) | jedes Wintersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

In diesem Modul wird ein grundlegendes Verständnis des Aufbaus und der Funktionsweise moderner Computer vermittelt. Dabei wird auf folgende Themengebiete eingegangen:

- Datendarstellung (ASCII, Unicode, b-adische Zahlen, Zweierkomplement, Qx.y Festkommazahlen, Gleitkommazahlen nach IEEE 754),
- Schaltalgebra (Definition nach Huntington, weitere Rechenregeln, vollständige Operatorensysteme, Wahrheitstabellen, disjunktive und konjunktive Normalform, KV-Diagramme, disjunktive und konjunktive Minimalform, Don't Care Belegungen),
- Digitale Logik (Gatter, De- und Encoder, Multiplexer und Demultiplexer, Shifter, Halb- und Volladdierer, Taktsignale, Flip-Flops und Latches (jeweils SR-, D-), zusammengesetzte Schaltungen mit diesen Elementen, Hazardfehler),
- Fehlererkennung und -korrektur (Verdopplung, Paritätsbit, Zweidimensionale Parität, Hammingdistanz, Hammingcode),
- Einführung von Java Bytecode als Beispiel für eine Assemblersprache (Übersetzung von einfachen Java Programmausschnitten in Java Byte Code, Entwurf einfacher Programmausschnitte direkt in Java Byte Code)
- Mikroarchitektur (Implementierung eines Prozessors für die Ausführung einer Teilmenge des Java Byte Codes, digitale Logik des Prozessors, Mikroprogramm für den Prozessor, Interpretation von Java Bytecode durch den Prozessor, Verbesserung der Architektur durch die Reduktion von Arbeitszyklen, Instruktions-Prefetching, Pipelining und Sprungvorhersage)
- Caching (Ersetzungsstrategien, Fully Associative Caches, Direct Mapped Caches, n-Way Set-Associative Caches) und
- Virtueller Speicher (Seitenersetzungsstrategien, Paging)

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- die verschiedenen Schichten einer Rechnerarchitektur unter Berücksichtigung ihrer Verbindungen untereinander beschreiben,
- boolesche Funktionen in verschiedenen Formen (Formel, Wahrheitstabelle, KV-Diagramm) angeben, auswerten, anhand des KV-Diagramms minimieren und die Äquivalenz von Funktionen zeigen,
- bewerten, ob eine gegebene Schaltung zu einer booleschen Funktion passt,
- Schaltungen aus bekannten Bauteilen analysieren und ihre Funktion beschreiben,
- Signalverlaufdiagramme zeichnen und anhand dieser bzw. anhand von KV-Diagrammen mögliche Hazardfehler in digitalen Schaltungen finden und beheben,
- erklären, wie eine CPU/ALU aus elementaren digitalen Schaltungen konstruiert wird und wie sie funktioniert,
- die Hammingdistanz von Wörtern und (endlicher) Codes berechnen und die Hammingdistanz bekannter Codes benennen,
- bekannte Verfahren zur Fehlererkennung und Korrektur anwenden und benennen welche Art von Fehlern durch diese erkannt bzw. behoben werden können,

- Programme für den besprochenen Beispielprozessor auf den Ebenen Binärcode, Mikrocode und Assemblercode analysieren und ihre Funktion beschreiben,
- Vor- und Nachteile der besprochenen Cachearten erörtern,
- Cacheverfahren auf Papier durchführen und dabei verschiedene Caches und Ersetzungsstrategien einander gegenüberstellen,
- erläutern, wofür Paging eingesetzt wird und Umrechnungen zwischen virtuellen und physischen Adressen vornehmen.

Literatur

- A. S. Tanenbaum, T. Austin: Structured Computer Organization. Pearson. Boston, 2013. 6th Edition
- Ergänzende Empfehlungen werden in der Lehrveranstaltung genannt.

Verwendbarkeit des Moduls

- Pflichtbereich Bachelor-Studiengang PO 2021
- Pflichtbereich Bachelor-Studiengang PO 2016, 2013
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik
- Modul CL6 im Bachelor-Studiengang Computerlinguistik

Teilnahmevoraussetzungen

- Inhaltlich: Es wird davon ausgegangen, dass die Studierenden parallel die Veranstaltung *Programmierung* hören oder über grundlegende Programmierkenntnisse verfügen.

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive und erfolgreiche Mitwirkung in den Übungen
- schriftliche Prüfung (Klausur, i. d. R. 90 Minuten)

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Stefan Conrad, Janine Golov, Prof. Dr. Martin Mauve

Theoretische Informatik (en. Theoretical Computer Science)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------------|------------------------------------|--------------------|
| 10 LP | 300 Stunden | 90 Stunden | 210 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (4 SWS) Übung (2 SWS) | jedes Sommersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

In diesem Modul werden die wichtigsten Grundlagen, Modelle, Methoden und Resultate der theoretischen Informatik eingeführt, insbesondere zur Theorie der formalen Sprachen und Automaten, zur Berechenbarkeitstheorie und zur Theorie der NP-Vollständigkeit.

Formale Sprachen und Automaten

- Grundbegriffe (Wörter, Sprachen und Grammatiken; die Chomsky-Hierarchie)
- Reguläre Sprachen (endliche Automaten; reguläre Ausdrücke; Gleichungssysteme; das Pumping-Lemma; Satz von Myhill und Nerode und Minimalautomaten; Abschlusseigenschaften und Charakterisierungen regulärer Sprachen)
- Kontextfreie Sprachen (Normalformen; das Pumping-Lemma; Abschlusseigenschaften kontextfreier Sprachen; der Algorithmus von Cocke, Younger und Kasami; Kellerautomaten)
- Deterministisch kontextfreie Sprachen (deterministische Kellerautomaten; Anwendung: Syntaxanalyse durch LL(k)-Parser)
- Kontextsensitive und L₀-Sprachen (Turingmaschinen; linear beschränkte Automaten; Zusammenfassung)

Berechenbarkeit

- Intuitiver Berechenbarkeitsbegriff und die These von Church
- Turing-Berechenbarkeit
- LOOP-, WHILE- und GOTO-Berechenbarkeit
- Primitiv rekursive und partiell rekursive Funktionen (primitiv rekursive Funktionen; die Ackermann-Funktion; allgemein und partiell rekursive Funktionen; der Hauptsatz der Berechenbarkeitstheorie)
- Entscheidbarkeit und Aufzählbarkeit
- Unentscheidbarkeit (der Satz von Rice; Reduzierbarkeit; das Postsche Korrespondenzproblem; Unentscheidbarkeit in der Chomsky-Hierarchie; Zusammenfassung)

NP-Vollständigkeit

- Probleme in P und NP (deterministische Polynomialzeit; das Erfüllbarkeitsproblem der Aussagenlogik; nichtdeterministische Polynomialzeit)
- NP-Vollständigkeit und der Satz von Cook

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- formale Sprachen in die Chomsky-Hierarchie einordnen,
- besprochene äquivalente Automatenmodelle ineinander bzw. in besprochene Grammatiken des entsprechenden Typs umformen,
- für eine gegebene Sprache eine sie erzeugende Grammatik bzw. einen sie akzeptierenden Automaten eines geeigneten Typs (z. B. endlicher Automat oder Kellerautomat oder lineare beschränkter Automat oder Turingmaschine) angeben,
- umgekehrt für eine gegebene Grammatik oder einen gegebenen Automaten die zugehörige Sprache

- bestimmen,
- Argumente für die Inäquivalenz von besprochenen Automatenmodellen bzw. Grammatiktypen geben,
 - den Aufbau eines Compilers angeben,
 - die Aufgaben und Methoden der lexikalischen und der Syntaxanalyse beschreiben,
 - die algorithmische Entscheidbarkeit von Problemen diskutieren,
 - Argumente für die Nichtentscheidbarkeit von Problemen geben,
 - argumentieren, dass es nicht berechenbare Funktionen gibt,
 - Fertigkeiten im Umgang mit formalen Begriffs- und Modellbildungen sowie mit formalen Argumentationsweisen sowie bestimmten Beweistechniken (wie etwa der Diagonalisierung) anwenden, und
 - Reduktionen zwischen Problemen angeben, um ihre Unentscheidbarkeit oder NP-Vollständigkeit zu zeigen.

Literatur

- Uwe Schöning: Theoretische Informatik kurz gefasst. Spektrum Akademischer Verlag. Heidelberg, 2008. 5. Auflage.
- John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman: Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie. Pearson Studium. München, 2002. 2. Auflage.
- Klaus W. Wagner: Theoretische Informatik. Eine kompakte Einführung. Springer-Verlag. Berlin, Heidelberg, 2003. 2. Auflage.
- Ergänzende Empfehlungen werden in der Lehrveranstaltung genannt.

Verwendbarkeit des Moduls

- Pflichtbereich Bachelor-Studiengang PO 2021
- Pflichtbereich Bachelor-Studiengang PO 2016, 2013
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik
- Modul CL6 im Bachelor-Studiengang Computerlinguistik

Teilnahmevoraussetzungen

- Inhalte des Moduls *Mathematik für Informatik 1* (alternativ: *Lineare Algebra I*)

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive und erfolgreiche Mitwirkung in den Übungen
- schriftliche Prüfung (Klausur, i. d. R. 90 Minuten)

Modulbeauftragte und hauptamtlich Lehrende

Mareike Mutz, Prof. Dr. Michael Leuschel, Prof. Dr. Jörg Rothe

Wissenschaftliches Arbeiten (en. Scientific Methods)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|---------------------|----------------------|---------------------------------|--------------------|
| 3 LP | 90 Stunden | 30 Stunden | 60 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) | jedes Wintersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Teil I: Praktische Methoden

- Literaturrecherche und effizientes Lesen der wissenschaftlichen Literatur
- Zitierung wissenschaftlicher Veröffentlichungen
- Struktur wissenschaftlicher Manuskripte und Protokolle
- Strukturelle Formatierung von Dokumenten
- Effektive Graphiken und Tabellen
- Effektive textliche und graphische Zusammenfassungen
- Der wissenschaftliche Publikationsprozess
- Präsentation wissenschaftlicher Ergebnisse
- Mündliche Präsentationen in Seminaren und auf Konferenzen
- Wissenschaftliche Poster
- Wissenschaftliche Kommunikation an ein nicht-wissenschaftliches Publikum (Outreach)
- Gute wissenschaftliche Praxis
- Vermeidung von Plagiaten
- „Best practice“ in der Wissenschaft
- Datenmanagement und Reproduzierbarkeit

Teil II: Ethische und gesellschaftliche Aspekte

- ethische Fragen und Herausforderungen in der Informatik
- gesellschaftliche, kulturelle und politische Bedeutung der Informatik

Teil III: Die wissenschaftliche Methode

- Falsifizierung und Beweis von Hypothesen
- Die Rolle von Hilfsannahmen in der Wissenschaft
- Paradigmen in der Wissenschaft
- Die „eiserne Regel“ der Wissenschaft: nur formale Beweise und Daten zählen
- Wissenschaftliche Veröffentlichungen als Erzählungen
- Der Unterschied zwischen Day Science (Ausführung) und Night Science (Kreativität)
- Die unterschiedlichen Sprachen von Day Science und Night Science
- Publikationsbias, p-hacking und Reproduzierbarkeit

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- die grundsätzlichen Methoden der Wissenschaft beschreiben, sowohl auf praktischer als auch auf abstrakter Ebene,
- den Prozess des wissenschaftlichen Publizierens und die zentrale Rolle von Zitierungen in der wissenschaftlichen Kommunikation beschreiben,
- den Peer-Review Prozess sowie die angemessene Struktur von Reviews beschreiben,
- wissenschaftliche Literatur identifizieren und lesen,
- angemessen strukturierte wissenschaftliche Texte und Grafiken entwerfen, gestalten und zusammenfassen

- ethische Fragen und Herausforderungen aus Sicht der Informatik identifizieren und die Herangehensweise an diese beschreiben,
- die gesellschaftliche, kulturelle und politische Bedeutung ihrer Fachdisziplin einschätzen,
- die Stellung von formalen Beweisen in der Informatik begründen,
- die zentrale Rolle von Hypothesen und ihrer Testung in der Wissenschaft erklären und
- die Rolle von Kreativität in der Wissenschaft würdigen und den Prozess des wissenschaftlichen Arbeitens von wissenschaftlicher Kommunikation unterscheiden.

Literatur

- Alan Chalmers: What is this thing called science? University of Queensland Press, Open University Press. 2013. 4th edition
- Michael Strevens: The Knowledge Machine: How Irrationality Created Modern Science. Liveright Publishing Corporation. 2021
- Itai Yanai & Martin Lercher: Night Science Collection. <https://www.biomedcentral.com/collections/night-science>
- Ergänzende Empfehlungen werden in der Lehrveranstaltung genannt.

Verwendbarkeit des Moduls

- Pflichtbereich Bachelor-Studiengang PO 2021
- Praxis- und Berufsorientierung für die Bachelor PO 2013 und PO 2016

Teilnahmevoraussetzungen

- keine

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive und erfolgreiche Mitwirkung in den Übungen
- Bestehen der schriftlichen Prüfung (Klausur, i. d. R. 90 Minuten)

Modulbeauftragte und hauptamtlich Lehrende

Mareike Mutz, Prof. Dr. Martin Lercher

Pflichtmodule der Mathematik (en. Compulsory Modules in Mathematics)

Im Bachelor-Studiengang Informatik an der HHU haben Sie zwei Möglichkeiten den Bereich Mathematik abzudecken. Sie können entweder den Pfad *Mathematik für Informatik* oder den Pfad *Mathematik der Mathematik* wählen. Die Module des Pfads *Mathematik für Informatik* finden Sie in diesem Modulhandbuch auf den kommenden Seiten. Die Module des Pfads *Mathematik der Mathematik* finden Sie im Modulhandbuch der Mathematik. Bitte entnehmen Sie diesem insbesondere die Inhalte, Lernziele und Voraussetzungen der Module. Folgende Module können im Bereich Mathematik gewählt werden:

- *Lineare Algebra I* (verpflichtend)
- *Analysis I* (verpflichtend)
- *Stochastik*
- *Numerik I*

Sie haben die Wahl, ob Sie *Stochastik* oder *Numerik I* belegen möchten. Beachten Sie jedoch, dass für die *Numerik I* werden weitere Fächer der Mathematik als Voraussetzungen empfohlen werden.

Mathematik für Informatik 1 (en. Mathematics for Computer Science 1)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|--|----------------------|------------------------------------|--------------------|
| 10 LP | 300 Stunden | 120 Stunden | 180 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (4 SWS) Übung (2 SWS) Tutorium (2 SWS) | jedes Wintersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

- Mathematische Grundlagen: Mengen, Relationen, Abbildungen, Abzählbarkeit und Gleichmächtigkeit, Aussagenlogik
- Reelle Zahlen: Körperaxiome und Anordnung der reellen Zahlen, Absolutbetrag, Induktionsbeweis, reelle Funktionen
- Der euklidische Raum: Lineare Gleichungen, Skalarprodukt, Lineare Abbildungen, Determinante
- Folgen, Vollständigkeit der reellen Zahlen, Reihen, Exponentialreihe und -funktion

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- die grundlegenden mathematischen Begriffe anwenden,
- mithilfe erlernter Methoden einfache und kurze Beweise schreiben,
- lineare Gleichungssysteme lösen,
- anhand des zentralen Grenzwertbegriffs der Analysis argumentieren und
- Übungsaufgaben selbstständig lösen und die Ergebnisse in der nötigen mathematischen Präzision darstellen.

Literatur

- B. Kreußler, G. Pfister: Mathematik für Informatiker. Springer. Berlin, Heidelberg, 2009.
- G. Teschl, S. Teschl: Mathematik für Informatiker I. Springer. Berlin, Heidelberg, 2013. 4. Auflage
- G. Teschl, S. Teschl: Mathematik für Informatiker II. Springer. Berlin, Heidelberg, 2014. 3. Auflage
- P. Hartmann: Mathematik für Informatiker. Springer, Wiesbaden, 2020. 7. Auflage
- W. Struckmann, D. Wätjen: Mathematik für Informatiker, Springer, Berlin, Heidelberg, 2016. 2. Auflage

Verwendbarkeit des Moduls

- Pflichtbereich Bachelor-Studiengang PO 2021

Teilnahmevoraussetzungen

- keine

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive und erfolgreiche Mitwirkung in den Übungsgruppen
- schriftliche Prüfung (Klausur, 120 Minuten)

Modulbeauftragte und hauptamtlich Lehrende

Dr. Nadja Hempel (Valentin), Dr. Andreas Rätz

Mathematik für Informatik 2 (en. Mathematics for Computer Science 2)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|--|----------------------|------------------------------------|--------------------|
| 10 LP | 300 Stunden | 120 Stunden | 180 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (4 SWS) Übung (2 SWS) Tutorium (2 SWS) | jedes Sommersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

- b-adische Brüche, Asymptotische Notation
- Differentialrechnung: Grenzwerte von Funktionen und Stetigkeit, elementare Funktionen, Maxima und Suprema, differenzierbare Funktionen, lokale Extrema und Mittelwertsätze, Regeln von de l'Hospital
- Integralrechnung: Riemannsches Integral, Hauptsatz der Differential- und Integralrechnung, uneigentliche Integrale
- Taylorentwicklung, Potenzreihen
- Algebraische Strukturen: Gruppen, Körper
- Lineare Algebra: Abstrakte Vektorräume, Unterräume, lineare Abbildungen, Basis, Dimension, Endomorphismen, Eigenwerte und Eigenvektoren

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- ausgewählte Techniken mit Anwendungen von Folgen und Reihen für die Informatik benutzen,
- weitere Grenzwertbegriffe für Funktionen anwenden,
- die Konzepte und Rechenregeln der Differential- und Integralrechnung sicher verwenden,
- asymptotisches Verhalten von Funktionen mit unterschiedlichen Techniken untersuchen,
- die grundlegenden Begriffe der linearen Algebra anwenden,
- mithilfe von linearen Gleichungssystemen Problemstellungen der Linearen Algebra lösen,
- die darstellende Matrix einer linearen Abbildung sowie die Übergangsmatrix zwischen zwei Basen berechnen.

Literatur

- B. Kreußler, G. Pfister: Mathematik für Informatiker. Springer. Berlin, Heidelberg, 2009.
- G. Teschl, S. Teschl: Mathematik für Informatiker I. Springer. Berlin, Heidelberg, 2013. 4. Auflage
- G. Teschl, S. Teschl: Mathematik für Informatiker II. Springer. Berlin, Heidelberg, 2014. 3. Auflage
- P. Hartmann: Mathematik für Informatiker. Springer, Wiesbaden, 2020. 7. Auflage
- W. Struckmann, D. Wätjen: Mathematik für Informatiker, Springer, Berlin, Heidelberg, 2016. 2. Auflage

Verwendbarkeit des Moduls

- Pflichtbereich Bachelor-Studiengang PO 2021

Teilnahmevoraussetzungen

- Inhaltlich: Inhalte des Moduls *Mathematik für Informatik 1*

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive und erfolgreiche Mitwirkung in den Übungsgruppen
- schriftliche Prüfung (Klausur, 120 Minuten)

Modulbeauftragte und hauptamtlich Lehrende

Dr. Nadja Hempel (Valentin), Dr. Andreas Rätz

Mathematik für Informatik 3 (en. Mathematics for Computer Science 3)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|--|----------------------|------------------------------------|--------------------|
| 10 LP | 300 Stunden | 120 Stunden | 180 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (4 SWS) Übung (2 SWS) Tutorium (2 SWS) | jedes Wintersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

- Endomorphismen: Eigenwerte und Eigenvektoren, Diagonalisbare Matrizen
- Codierungstheorie: Hamming-Metrik, lineare Codes, perfekte Codes
- mehrdimensionale Analysis: Differentialrechnung in mehreren Variablen, Extremwertaufgaben für Funktionen mit mehreren Variablen
- Numerik: numerische Verfahren für lineare Gleichungssysteme, Interpolation, Behandlung nichtlinearer Gleichungen (Newton-Verfahren)
- Stochastik: Ereignisraum, Wahrscheinlichkeit, bedingte Wahrscheinlichkeit, Zufallsvariable (diskret und stetig), Verteilungsfunktion, Erwartungswert, Standardabweichung, Varianz, Binomialverteilung, Poisson-Verteilung, Exponentialverteilung, Gleichverteilung, Normalverteilung

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- Berechnung von Eigenwerten und Eigenvektoren von Matrizen und Endomorphism sowie Diagonalgestalt diagonalisierbarer Endomorphism/Matrizen.
- die Differentialrechnung für Funktionen mit mehreren reellen Variablen auf konkrete Problemstellungen anwenden.
- die behandelten numerische Techniken im Hinblick auf Effizienz und Umsetzbarkeit bewerten und
- die behandelten Begriffe der Stochastik beschreiben sowie die behandelten Verteilungen kennen und zur Anwendung bringen.

Literatur

- B. Kreuzler, G. Pfister: Mathematik für Informatiker. Springer. Berlin, Heidelberg, 2009.
- G. Teschl, S. Teschl: Mathematik für Informatiker I. Springer. Berlin, Heidelberg, 2013. 4. Auflage
- G. Teschl, S. Teschl: Mathematik für Informatiker II. Springer. Berlin, Heidelberg, 2014. 3. Auflage
- P. Hartmann: Mathematik für Informatiker. Springer, Wiesbaden, 2020. 7. Auflage
- W. Struckmann, D. Wätjen: Mathematik für Informatiker, Springer, Berlin, Heidelberg, 2016. 2. Auflage

Verwendbarkeit des Moduls

- Pflichtbereich Bachelor-Studiengang PO 2021

Teilnahmevoraussetzungen

- Inhaltlich: Inhalte der Module *Mathematik für Informatik 1 und 2*

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive und erfolgreiche Mitwirkung in den Übungsgruppen
- schriftliche Prüfung (Klausur, 120 Minuten)

Modulbeauftragte und hauptamtlich Lehrende

Dr. Nadja Hempel (Valentin), Dr. Andreas Rätz

Lehreinheiten für den Wahlbereich (en. Courses for elective areas)

Der Wahlbereich umfasst 45 Leistungspunkte. Für die Zusammensetzung der Module gilt:

- Zwecks verbesserter Studierbarkeit des Studiengangs und zur Erhöhung der Wahlmöglichkeiten der Studierenden werden im Bachelor-Studiengang Informatik Module in unterschiedlichen Größen angeboten. Module müssen so kombiniert werden, dass die Gesamtsumme der LP den Erfordernissen der Prüfungsordnung entspricht. Für den Wahlpflichtbereich können Module frei kombiniert werden.
- 20 LP dürfen aus dem Lehrangebot anderer Fächer stammen. Auf den [Webseiten des Studiengangs](#) wird eine Liste gepflegt, welcher entnommen werden kann, welche Fächer anerkannt werden.
- 5 LP dürfen über Angebote außerhalb der Fakultäten abgedeckt werden. Diese gehen nicht in die Gesamtnote des Bachelorabschlusses ein. Zu den entsprechenden Angeboten zählen insbesondere die Angebote der Studierendenakademie (Career, Soft & Study Skills, Sprachen).

Im Bachelor-Studiengang Informatik können bis zu 30 LP als Zusatzleistungen erbracht werden. Diese können auch aus dem Lehrangebot des Master-Studiengangs Informatik stammen. Die angebotenen Module entnehmen Sie bitte dem Modulhandbuch für den Master-Studiengang, welches Sie auf den Webseiten der Informatik finden können. Die Teilnahme an diesen Modulen ist nur zulässig, wenn die Veranstaltungen Programmierung, Rechnerarchitektur, Algorithmen und Datenstrukturen sowie Theoretische Informatik bestanden sind.

Ein Modul für den Bachelor-Studiengang kann nur dann für die Individuelle Ergänzung im Master-Studiengang Informatik verwendet werden, wenn es nicht bereits für ein vorangegangenes Bachelor-Studium verwendet wurde und dafür zugelassen ist (dies ist seit dem 01.09.2022 in der Regel nicht mehr der Fall).

Weiterhin sind folgende Module des [Master AI and Data Science](#) zulässig:

| Titel | Verantwortliche Personen | Turnus | LP | Sprache |
|------------------|----------------------------|----------------------|-------|---------|
| Machine Learning | Prof. Dr. Stefan Harmeling | jedes Wintersemester | 10 LP | EN |

Algorithmen in der Bioinformatik (en. Algorithms in Bioinformatics)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) | jedes Sommersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Diese Modul vermittelt einführende Konzepte der algorithmischen Bioinformatik. Die Studierenden vertiefen und wenden diese in theoretischen und praktischen Übungen an.

- Biologische Grundlagen
- Exhaustive Suche: DNA-Motive
- Gierige Algorithmen: Genom-Umordnungen
- Dynamische Programmierung: Sequenz-Alignment
- Graphen-Algorithmen: Assembly
- Kombinatorische Mustersuche und Suffixbäume
- Clustering
- Phylogenetische Bäume und molekulare Evolution
- Hidden Markov Modelle: CpG-Inseln

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- die besprochenen algorithmische Designprinzipien anwenden, Korrektheit von Algorithmen beweisen und Laufzeiten analysieren,
- zwischen verschiedenen Klassen von Algorithmen unterscheiden und die Konsequenzen erklären,,
- klassische Bioinformatikalgorithmen erklären und anwenden,
- viele dieser Algorithmen in Python implementieren und
- für eine bestimmte Aufgabe geeignete Algorithmen auswählen.

Literatur

- Neil C. Jones, Pavel A. Pevzner: An Introduction to Bioinformatics Algorithms. The MIT Press, 2004
- Phillip Compeau, Pavel A. Pevzner : Bioinformatics Algorithms, An Active Learning Approach, Vol. I and II, Active Learning Publishers, 2015

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

- Inhaltlich: Inhalte der Module *Programmierung* und *Algorithmen und Datenstrukturen* und *Mathematik*

für Informatik 1 (oder Lineare Algebra I oder Analysis I)

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive Teilnahme an den Übungen
- erfolgreiches Bearbeiten der Übungsaufgaben (50%)
- abschließende Prüfung (schriftlich, i. d. R. 90 Minuten)

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Gunnar W. Klau

Algorithmische Geometrie (en. Computational Geometry)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) | unregelmäßig | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Dieses Modul vermittelt grundlegende Kenntnisse aus folgenden Bereichen:

- konvexe Hülle in der Ebene
- Plane-Sweep Algorithmen / Segment-Schnittprobleme- Distanzprobleme- Geometrisches Divide and Conquer / dichtestes Punktpaar
- Voronoi Diagramme / Delaunay-Triangulierungen / nächster Nachbar Anfragen
- Triangulierung von Polygonen / monotonen Polygonen
- Bereichsanfragen / Ham-Sandwich Theorem
- Rechteckabfragen / Bereichsbäume
- Punkt/Gerade-Dualität / Geradenarrangement
- kleinste umschließende Kreise / randomisierte Algorithmen

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- mit Datenstrukturen für die Berechnung der konvexe Hülle, eines Voronoi-Diagramms, einer Delauney-Triangulierung oder eines Graden-Arrangement umgehen
- Plane-Sweep Techniken, Teile-und-herrsche Verfahren und randomisierte Ansätze für das Lösen von geometrischen Problemen einsetzen
- untere Schranken für die Komplexität geometrischer Fragestellungen auf Punktmengen in der Euklidischen Ebene bestimmen

Literatur

- de Berg et al: Computational Geometry, Algorithms and Applications. Springer-Verlag. Berlin, 2000. 2. rev. ed.
- Preparata, Shamos: Computational Geometry, an Introduction. Springer-Verlag. New York, 1985.
- Edelsbrunner: Algorithms in Combinatorial Geometry, EATCS Monographs in Computer Science 10. Springer-Verlag, 1987.
- Matousek: Lectures on Discrete Geometry, Graduate Texts in Mathematics, 212. Springer-Verlag, New York, 2002.

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive Mitarbeit in den Übungen,
- Abgabe ausgewählter Hausaufgaben,
- schriftliche Klausur (i. d. R. 90 Minuten) oder mündliche Prüfung am Ende des Semesters

Modulbeauftragte und hauptamtlich Lehrende

Prof Dr. Egon Wanke

Algorithmische Komplexitätstheorie (en. Computational Complexity Theory)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) | unregelmäßig | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Dieses Modul vermittelt grundlegende Kenntnisse aus folgenden Bereichen:

- Einführung in die algorithmische Komplexität (Nichtdeterminismus, NP-Vollständigkeit, ...)
- Sat, 3-SAT, Clique, IS, VC, 3-DM, Dominating Set, 3-Partition
- Pseudopolynomielle Algorithmen, Rucksack-, Partitions-Probleme
- Approximationsalgorithmen
- Schaltkreiskomplexität
- Parameterisierte Algorithmen, FPT
- Satz von Savich, Satz von Immerman und Szelepcsényi
- Randomisierte Algorithmen

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- nichtdeterministische Berechnungsmodelle anwenden,
- die Komplexität algorithmischer Probleme abschätzen,
- Grundlegende Techniken zur Approximation von Lösungen anwenden,
- pseudo-polynomielle Lösungsansätze entwickeln, und
- parametrisierte Fragestellungen analysieren.

Literatur

- Michael R. Garey and David S. Johnson: Computers & Intractability: A Guide to the Theory of NP-Completeness. Freeman. 1979
- Christos H. Papadimitriou: Computational Complexity. Addison-Wesley. 2008

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

- Inhaltlich: Inhalte der Module *Algorithmen und Datenstrukturen* und *Mathematik für Informatik 1* (oder *Lineare Algebra I* und *Analysis I*)

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive Mitarbeit in den Übungen,
- Abgabe ausgewählter Hausaufgaben,
- schriftliche Klausur (i. d. R. 90 Minuten) oder mündliche Prüfung am Ende des Semesters

Modulbeauftragte und hauptamtlich Lehrende

Prof Dr. Egon Wanke

Angewandte Algorithmik (en. Applied Algorithmics)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) | jedes Wintersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

„In theory, there is no difference between theory and practice. In practice, there is.“

Algorithmen bilden die Grundlage jedes Computerprogramms. Traditionell liegt der Fokus des Algorithmen-Designs auf der Theorie effizienter Algorithmen und deren worst-case-Analyse. In dieser Veranstaltung geht es jedoch um praktisch effiziente Algorithmen für beweisbar schwere Optimierungsprobleme. Das Ziel ist dabei, die Prinzipien der Optimalität nicht (vollständig) aufzugeben. Themengebiete sind:

- Wesentliche Aspekte der Algorithmik und Komplexität
- Vollständige Enumeration und Dynamische Programmierung
- Branch-and-Bound
- Approximationsalgorithmen
- Heuristiken und Metaheuristiken
- (Ganzzahlige) Lineare Programmierung
- Festparameter-Algorithmik
- Modellierungen mit Satisfiability (SAT)

Viele der Themengebiete werden nur einführend behandelt. Diese können dann in Spezialmodulen oder im Masterstudium vertieft werden.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- vorgestellte Techniken der Angewandten Algorithmik auf neue Probleme anwenden,
- Probleme praktisch effizient lösen, diese Lösungen implementieren, testen und evaluieren.

Literatur

- Steven Skiena, The Algorithm Design Manual, Second Edition, Springer, 2008

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

- Inhaltlich: Inhalte der Module *Programmierung, Algorithmen und Datenstrukturen, Theoretische Informatik* und *Mathematik für Informatik 1* (oder *Lineare Algebra I* oder *Analysis I*)

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive Teilnahme an den Übungen
- erfolgreiches Bearbeiten der Übungsaufgaben (50%)
- abschließende Prüfung (schriftlich, i. d. R. 90 Minuten)

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Gunnar W. Klau

Betriebssysteme und Systemprogrammierung (en. Operating Systems and System Programming)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------------|------------------------------------|--------------------|
| 10 LP | 300 Stunden | 90 Stunden | 210 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (4 SWS) Übung (2 SWS) | jedes Wintersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

- C Programmierung
- Bibliotheken, Binder und Lader
- Prozesse und Threads
- Scheduling: verschiedene Verfahren (Multilevel, Feedback, Echtzeit), Fallstudien: Linux, Windows, ...
- Synchronisierung: Mutex, Semaphore, Verklemmungen, sperrfreie Synchronisierung
- Hauptspeicher: Heap, Stack, Speicherverwaltung, Garbage Collection
- Virtueller Speicher: ein und mehrstufig, invertierte Seitentabellen, Auslagerungsstrategien, Speicherverwaltung in Linux
- Sekundärspeicher: Aufbau einer Festplatte/SSD, Partitionen, Speicherverwaltung
- Dateisysteme: FAT, UNIX, ext4, NTFS (mit Journaling)
- Interprozesskommunikation: Signale, Message Queue, Pipes, Shared-Memory, Sockets
- Ein- und Ausgabe: Interrupts, E-/A-Software, Linux Kernel Module und Gerätetreiber
- Sicherheit: Zugriffskontrolle, HW-Schutz, Angriff durch Pufferüberlauf, Meltdown, Address Space Layout Randomization, Kernel Page Table Isolation
- Architekturen: Monolithen, Mikrokerne, virtuelle Maschinen, Client/Server

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- das Zusammenspiel von Betriebssystem-Kern, Treiber und Hardware in eigenen Worten beschreiben
- Betriebssystemkonzepte in eigenen Worten erklären und miteinander vergleichen
- Systemprogramme in der Sprache C auf Basis der Systemaufrufchnittstelle UNIX entwickeln
- einfache parallele Programme mit mehreren Threads und geeigneten Synchronisierungslösungen konzipieren und programmieren
- Sicherheitsprobleme und Gegenmaßnahmen durch das Betriebssystem und den Hardwareschutz (am Beispiel der x86 Architektur) erklären

Literatur

- A. Tanenbaum: „Modern Operating Systems“, 4. Aufl., Prentice Hall, 2014.
- W. Stallings, „Operating Systems: Internals and Design Principles“, Prentice Hall, 9. Aufl., 2017.

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

- Inhalte der Module *Programmierung* und *Rechnerarchitektur*

Voraussetzungen für die Vergabe von Leistungspunkten

- Erfolgreiche Teilnahme an der Prüfung am Ende der Veranstaltung.
- Die Klausur besteht aus zwei Teilen. 50% praktische Programmieraufgaben am PC und 50% Aufgaben auf Papier

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Michael Schöttner

Competitive Programming A (en. Competitive Programming A)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|--|----------------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) praktische Übung (2 SWS) | jedes Wintersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Competitive Programming ist ein Sport, bei dem es darum geht, Programmierprobleme möglichst schnell und unter vorgegebenen Zeit- und Platzvorgaben zu lösen. Dieses Modul vermittelt grundlegende Techniken, um verschiedene Arten von Problemen zu erkennen und in C++ zu lösen. Ziel ist es, die Teilnehmer dieses Moduls auf die Teilnahme an nationalen und internationalen Programmierwettbewerben (z. B. ICPC) vorzubereiten. Während des Semesters wird es neben wöchentlichen Programmierhausaufgaben auch einige Live-Contests während der Vorlesungszeit geben.

Themen des Moduls:

- Basisalgorithmen (Binärsuche, Sortieren, Dynamisches Programmieren, Backtracking, Prefixsummen)
- Greedy Algorithmen
- Graphenprobleme (DFS, kürzeste Wege, Spannbäume, Union-Find)
- Segmentbäume

Weitere Themen werden im Modul Competitive Programming B im Sommersemester behandelt. Die beiden Module ergänzen sich und können unabhängig voneinander belegt werden.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden

- Datenstrukturen und Algorithmen der C++ Standard Library benutzen,
- die Laufzeit von Programmen für gegebene Eingabegrößen abschätzen,
- gängige Lösungsmuster in einfachen Aufgaben von Programmierwettbewerben zu den Themen des Moduls erkennen, und
- effiziente Programme für einfache Aufgaben von Programmierwettbewerben zu den Themen des Moduls entwerfen und implementieren.

Literatur

- Ahmed Shamsul Arefin: Art of Programming Contest. Second Edition, Special Online Edition for UVa Online Judge Users
- Steven Halim, Felix Halim, and Suhendry Effendy: Competitive Programming 4: The Lower Bound of Programming Contest in the 2020s, Books 1+2. Lulu Press. 2020
- Nicolai Josuttis: The C++ Standard Library - A Tutorial and Reference. Addison Wesley Longman. 1999. 2nd edition
- Steven Skiena: The Algorithm Design Manual. Springer. 2008. 2nd edition.
- Steven Skiena and Miguel Revilla: Programming Challenges - The Programming Contest Training Manual. Springer Verlag. 2003.

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

- Inhaltlich: Gute Programmierkenntnisse in C++, Java oder Python und Grundkenntnisse von Datenstrukturen und Algorithmen.

Voraussetzungen für die Vergabe von Leistungspunkten

- 50% aus den wöchentlichen Hausaufgaben, mindestens 4 Wertungen sind erforderlich
- 50% aus dem Abschneiden bei Live-Contests, mindestens 4 Wertungen sind erforderlich

Modulbeauftragte und hauptamtlich Lehrende

apl. Prof. Dr. Rudolf Fleischer.

Competitive Programming B (en. Competitive Programming B)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|--|----------------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) praktische Übung (2 SWS) | jedes Sommersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Competitive Programming ist ein Sport, bei dem es darum geht, Programmierprobleme möglichst schnell und unter vorgegebenen Zeit- und Platzvorgaben zu lösen. Dieses Modul vermittelt grundlegende Techniken, um verschiedene Arten von Problemen zu erkennen und in C++ zu lösen. Ziel ist es, die Teilnehmer dieses Moduls auf die Teilnahme an nationalen und internationalen Programmierwettbewerben (z. B. ICPC) vorzubereiten. Während des Semesters wird es neben wöchentlichen Programmierhausaufgaben auch einige Live-Contests während der Vorlesungszeit geben.

Themen des Moduls:

- Geometrische Probleme (konvexe Hülle, Schnittbildung)
- Zahlentheoretische Probleme (Primzahlzerlegung, Teilbarkeitsprobleme)
- Wortprobleme (Teilwörter)

Weitere Themen werden im Modul Competitive Programming A im Wintersemester behandelt. Die beiden Module ergänzen sich und können unabhängig voneinander belegt werden.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden

- Datenstrukturen und Algorithmen der C++ Standard Library benutzen,
- die Laufzeit von Programmen für gegebene Eingabegrößen abschätzen,
- gängige Lösungsmuster in einfachen Aufgaben von Programmierwettbewerben zu den Themen des Moduls erkennen, und
- effiziente Programme für einfache Aufgaben von Programmierwettbewerben zu den Themen des Moduls entwerfen und implementieren.

Literatur

- Ahmed Shamsul Arefin: Art of Programming Contest. Second Edition, Special Online Edition for UVa Online Judge Users
- Steven Halim, Felix Halim, and Suhendry Effendy: Competitive Programming 4: The Lower Bound of Programming Contest in the 2020s, Books 1+2. Lulu Press. 2020
- Nicolai Josuttis: The C++ Standard Library - A Tutorial and Reference. Addison Wesley Longman. 1999. 2nd edition
- Steven Skiena: The Algorithm Design Manual. Springer. 2008. 2nd edition.
- Steven Skiena and Miguel Revilla: Programming Challenges - The Programming Contest Training Manual. Springer Verlag. 2003.

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

- Inhaltlich: Gute Programmierkenntnisse in C++, Java oder Python und Grundkenntnisse von Datenstrukturen und Algorithmen.

Voraussetzungen für die Vergabe von Leistungspunkten

- 50% aus den wöchentlichen Hausaufgaben, mindestens 4 Wertungen sind erforderlich
- 50% aus dem Abschneiden bei Live-Contests, mindestens 4 Wertungen sind erforderlich

Modulbeauftragte und hauptamtlich Lehrende

apl. Prof. Dr. Rudolf Fleischer.

Compilerbau (en. Compiler Construction)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|---|----------------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 90 Stunden | 60 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) Praktische Übung (2 SWS) | jedes Wintersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

In der Veranstaltung werden die Grundlagen des Compilerbaus besprochen.

- Lexikale Analyse (reguläre Ausdrücke und endliche Automaten)
- Syntaxanalyse (kontextfreie Grammatiken und deterministisches Parsing)
- Semantische Analyse
- Code-Generierung
- Benutzen von Werkzeugen zur automatischen Erzeugung von Compilern

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- beschreiben wie Programmiersprachen übersetzt und implementiert werden,
- Syntaxbeschreibungen einer Programmiersprache beschreiben und anpassen (insbesondere ermitteln ob die Beschreibung für eine automatisierte Bearbeitung in einem Compiler geeignet ist) und
- einen Parser, bzw. Compiler für eine neue Programmiersprache entwickeln.

Literatur

- Andrew W. Appel: Modern Compiler Implementation in Java. Cambridge University Press. 2nd Edition
- Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman: Compilers: Principles, Techniques, and Tools. Addison Wesley. 2nd Edition

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

- Inhaltlich: Inhalte des Moduls *Programmierung*

Voraussetzungen für die Vergabe von Leistungspunkten

- Erfolgreiche Bearbeitung der Pflichtübungen
- Erfolgreiche Entwicklung eines eigenen Compilers
- Bestehen der Klausur

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Michael Leuschel, Dr. John Witulski

Computer-gestützte Beweisführung (en. Computer-aided reasoning)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 45 Stunden | 105 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (1 SWS) | unregelmäßig | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

In dieser fachübergreifenden Veranstaltung geht es um Beweisassistenten: Computerprogramme, mit denen man mathematische Beweise formal konstruieren und verifizieren kann. Wir sprechen über die theoretischen und technischen Grundlagen von Beweisassistenten am Beispiel des Beweisassistenten Lean: Funktionale Programmierung, Typentheorie und Beweistaktiken. Schließlich schauen wir uns an, wie Beweisassistenten für mathematische, informatische und linguistische Anwendungen eingesetzt werden können.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- Typisierungsregeln von Typentheorien erklären und anwenden,
- Datentypen, mathematische Aussagen und ihre Beweise im Calculus of Inductive Constructions formalisieren und
- diese theoretischen Kenntnisse praktisch im Beweisassistenten Lean umsetzen.

Literatur

- Smolka: Modeling and Proving in Computational Type Theory Using the Coq Proof Assistant. https://www.ps.uni-saarland.de/~smolka/drafts/icl_book.pdf
- Avigad, de Moura, Kong, Ullrich: Theorem Proving in Lean 4. https://leanprover.github.io/theorem_proving_in_lean4/
- Eigenes Skript

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflichtbereich Bachelor-Studiengang PO 2013 und PO 2016

Teilnahmevoraussetzungen

- Inhaltlich: Inhalte der Module *Mathematik für Informatik 1*, *Programmierung* und *Theoretische Informatik*

Voraussetzungen für die Vergabe von Leistungspunkten

- Aktive und erfolgreiche Teilnahme an den Übungsgruppen
- Bestehen der schriftlichen Klausur

Modulbeauftragte und hauptamtlich Lehrende

Dr. Alexander Bentkamp

Data Science 2 (en. Data Science 2)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|--------------------------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) | jedes zweite Wintersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Data Science ist die Anwendung statistischer Methoden und Methoden des maschinellen Lernens auf Daten jedweder Art mit Hilfe eines Rechners zur Modellierung von Systemen und Vorhersage von Verhalten. Im Modul Data Science 2 werden die Grundlagen aus Data Science anwendungsnah vertieft, vor Allem hinsichtlich der Themengebiete:

- Große Datenmengen (Big Data)
- Datenparallelverarbeitung (Algorithmen und Softwarepakete)

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- Algorithmen für große Datenmengen und/oder Datenströme analysieren mit Bezug auf die Laufzeit- und Kommunikationskomplexität,
- insbesondere in den Themenbereichen Nachbarsuche in hochdimensionalen Daten, Locality Sensitive Hashing (LSH), Dimensionalitätsreduktion, Empfehlungssysteme, Clustering (Varianten von k-means), Linkanalyse (PageRank), Werbung im Web,
- entscheiden, für welche Art und Umfang von Daten Algorithmen ohne Parallelisierung ungeeignet sind,
- welche Techniken der Parallelverarbeitung für gegebene Probleme geeignet sind,
- insbesondere map-reduce,
- und diese implementieren bzw. vorhandene Softwarepakete geeignet einsetzen.

Literatur

- Leskovec, Rajaraman and Ullman: Mining of Massive Datasets. Cambridge University Press. 2020. 3rd Edition (mmds.org)

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016

Teilnahmevoraussetzungen

- Inhaltlich: Inhalte der Module *Data Science* oder *Machine Learning*

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive und erfolgreiche Mitwirkung in den theoretischen und praktischen Übungen
- Bestehen der schriftlichen Prüfung (Klausur, i. d. R. 90 Minuten)

Modulbeauftragte und hauptamtlich Lehrende

Dr. Konrad Völkel

Datenbanken: Weiterführende Konzepte (en. Databases: Further Concepts)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|--|----------------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) praktische Übung (2 SWS) | jedes Wintersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Aufbauend auf den Grundlagen des Datenbankentwurfs und der Anfragesprachen für relationale Datenbanksysteme werden in diesem Modul weitere Datenbankkonzepte vorgestellt. Neben den Grundlagen steht vor allem die praktische Umsetzung (Datenbankentwurf und -anwendungsprogrammierung) im Vordergrund. Die folgenden Themen werden dafür in der Vorlesung behandelt:

- Datenbankdefinition (in SQL)
- Datenbankanwendungsprogrammierung und Datenbankanbindung im Web
- Datenschutz und Datensicherheit (SQL Injection; Sichten und Rechte in Datenbanken)
- Trigger
- Implementierungsaspekte (physische Speicherung; Indexstrukturen)
- Algorithmen für Anfrageoperatoren und Anfrageoptimierung

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- selbstständig Datenbankanwendungen entwickeln (einschl. Datenbankentwurf, Datenbankdefinition und Anwendungsprogrammierung),
- grundlegende Aspekte des Datenschutzes und der Datensicherheit in der Entwicklung von Datenbankanwendungen berücksichtigen und
- zentrale Implementierungskonzepte zur Speicherung von Daten und elementare Datenstrukturen und Algorithmen zur Anfragebearbeitung erklären und bewerten.

Literatur

- G. Saake, K.-U. Sattler, A. Heuer: Datenbanken – Konzepte und Sprachen. mitp Verlag. 2018. TODO Auflage
- G. Saake, K.-U. Sattler, A. Heuer: Datenbanken – Implementierungstechniken. mitp Verlag. 2011. 3. Auflage
- Kemper, A. Eickler: Datenbanksysteme – Eine Einführung. Oldenbourg Verlag. 2015. 10. Auflage
- R. Elmasri, S.B. Navathe: Fundamentals of Database Systems. Pearson. 2016. 7th edition
- H. Garcia-Molina, J.D. Ullman, J. Widom: Database Systems: The Complete Book. Pearson. 2009. 2nd edition

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete

Teilnahmevoraussetzungen

- Inhaltlich: Inhalte des Moduls *Datenbanken: Eine Einführung*

Voraussetzungen für die Vergabe von Leistungspunkten

- Aktive und erfolgreiche Teilnahme an den Übungen (praktische, aufeinander aufbauende Aufgaben)
- abschließende Hausarbeit (Projekt zur Datenbankentwicklung)

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Stefan Conrad

Datenvisualisierung (en. Data Visualization)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) | unregelmäßig | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Es werden Kenntnisse im Bereich Datenvisualisierung vermittelt.

- Menschliche Wahrnehmung, Ästhetik, Koordinatensysteme
- Farbskalen, Visualisierungen ausgewählter Datenarten (quantitative Werte, Verteilungen, Proportionen, Trends, Geodaten, Ungenauigkeiten), Beschriftung, Tabellen
- Visualisierung Multidimensionaler Daten, Storytelling, Bildformate und Visualisierungswerkzeuge

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden

- geeignete Visualisierung für die entsprechenden Datentypen auswählen und erstellen,
- bei der Erstellung von Visualisierungen das erworbene Wissen über menschliche Wahrnehmung berücksichtigen,
- geeignete Bildformate und Tools zur Visualisierung auswählen.

Literatur

- Ergänzende Empfehlungen werden in der Lehrveranstaltung genannt.
- Wilke, Claus O.: Datenvisualisierung – Grundlagen und Praxis: Wie Sie aussagekräftige Diagramme und Grafiken gestalten. Sebastopol: O'Reilly, 2020.

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

- keine

Voraussetzungen für die Vergabe von Leistungspunkten

- Gruppenarbeit mit Präsentation; zu 20% gewichtet
- Ausarbeitung zur Gruppenarbeit am Ende des Semester; zu 80% gewichtet

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Dominik Heider, Dr. Hannah Franziska Löchel

Datenvisualisierung (en. Data Visualization)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) | unregelmäßig | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Es werden Kenntnisse im Bereich Datenvisualisierung vermittelt.

- Menschliche Wahrnehmung, Ästhetik, Koordinatensysteme
- Farbskalen, Visualisierungen ausgewählter Datenarten (quantitative Werte, Verteilungen, Proportionen, Trends, Geodaten, Ungenauigkeiten), Beschriftung, Tabellen
- Visualisierung Multidimensionaler Daten, Storytelling, Bildformate und Visualisierungswerkzeuge

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden

- geeignete Visualisierung für die entsprechenden Datentypen auswählen und erstellen,
- bei der Erstellung von Visualisierungen das erworbene Wissen über menschliche Wahrnehmung berücksichtigen,
- geeignete Bildformate und Tools zur Visualisierung auswählen.

Literatur

- Ergänzende Empfehlungen werden in der Lehrveranstaltung genannt.
- Wilke, Claus O.: Datenvisualisierung – Grundlagen und Praxis: Wie Sie aussagekräftige Diagramme und Grafiken gestalten. Sebastopol: O'Reilly, 2020.

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

- keine

Voraussetzungen für die Vergabe von Leistungspunkten

- Aktive und erfolgreiche Mitwirkung in den Übungen
- Prüfung zur Vorlesung und Übungen am Ende des Semesters

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Dominik Heider, Dr. Hannah Franziska Löchel

Digitale Innovation: Von der Idee zur Wirkung (en. Digital Innovation: From Idea to Impact)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|-------------------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Häufigkeit des Angebots | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) | jedes Sommersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Dieser Kurs richtet sich an künftige GründerInnen, CTOs (Chief Technology Officers) und alle, die wissen wollen, wie man technologiebasierte Innovationen entwickelt und in Start-ups und großen Unternehmen auf den Markt bringt.

Mittels Vorlesung und praktischen Anwendungen werden die Prozesse, das Wissen und die Techniken vermittelt, die erforderlich sind, um aus einer Idee ein konkretes marktfähiges Produkt zu machen mit Anziehungskraft auf die Zielgruppe und nachhaltiger Wirkung auf Wirtschaft und Gesellschaft.

Im Rahmen von Gruppenprojekten wenden die Studierenden die Werkzeuge und Methoden an, um eigene digitale Ideen zu kreieren, zu bewerten, in ein digitales Produkt zu übersetzen und prototypisch zu erstellen.

Es werden Beispiele und Fallstudien aus verschiedenen Bereichen behandelt.

Gastvortragende aus dem regionalem und/oder nationalem Innovations-Ökosystem teilen ihre Erfahrungen.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- digitale Innovationen kreieren, entwickeln, analysieren und bewerten,
- digitale Geschäftsmodelle und ihre Komponenten modellieren, analysieren und diskutieren,
- digitale Innovation in Prototypen umsetzen und bewerten,
- Gruppenergebnisse vor Gleichaltrigen und Experten präsentieren und bewerten und
- Feedback geben.

Literatur

- Allmendinger, M.; Horstmann, M.; Horstmann, O.: Digitale Innovationen entwickeln. Die besten Ansätze und Methoden. Haufe Lexware GmbH, 2020.
- Osterwalder, A; Pigneur, Y.: Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. John Wiley & Sons, 2010. 1st edition
- Tidd, J; Bessant, J. R.: Managing Innovation: Integrating Technological, Market and Organizational Change. Wiley, 2020. 7th edition
- Ergänzende Empfehlungen werden in der Lehrveranstaltung genannt.

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016

- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

keine

Voraussetzungen für die Vergabe von Leistungspunkten

In der Regel

- Vortrag
- schriftliche Ausarbeitung

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Steffi Haag

Einführung in algorithmische Spieltheorie (en. Introduction to Algorithmic Game Theory)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) | jedes Sommersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

In diesem Modul werden die wichtigsten Grundlagen aus dem Gebiet der algorithmischen Spieltheorie in der Breite präsentiert. Dazu werden verschiedene Felder innerhalb der algorithmischen Spieltheorie vorgestellt und besonders wichtige Ergebnisse vermittelt.

- Strategische Spiele (Normalform, Dominanz, reine und gemischte Gleichgewichte und ihre Existenz)
- Potentialspiele (Atomic- and Non-atomic selfish routing games, Preis der Stabilität und Anarchy, Potentialfunktionen)
- Mechanism Design (einfache Auktionen, Myerson's Lemma, VCG-Preise, Revenue Maximization)
- Mechanism Design without Money (Kidney Exchange, Stable Matching)

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- die besprochenen spieltheoretischen Modelle erkennen und erklären,
- die besprochenen Resultate wiedergeben und anwenden,
- die besprochenen Eigenschaften und Metriken erklären und an einfachen Anwendungen analysieren und
- neue spieltheoretische Modelle oder Mechanismen entwickeln.

Literatur

- Noam Nisan, Tim Roughgarden, Éva Tardos, Vijay V. Vazirani: Algorithmic Game Theory. Cambridge University Press. 2007.

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

- Inhaltlich: Inhalte der Module *Algorithmen und Datenstrukturen* und *Mathematik für Informatik 1* (alternativ: *Lineare Algebra I* oder *Analysis I*)

Voraussetzungen für die Vergabe von Leistungspunkten

- erfolgreiche Bearbeitung der Übungen
- schriftliche Klausur (i. d. R. 90 Minuten) oder mündliche Prüfung am Ende des Semesters

Modulbeauftragte und hauptamtlich Lehrende

Dr. Andreas Abels

Einführung in Deep Learning (en. Introduction to Deep Learning)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|--------------------------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) | jedes zweite Wintersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Aufbauend auf den Grundkenntnissen des maschinellen Lernens aus dem *Data Science* Modul widmen wir uns einer Auswahl an Kernthemen der Theorie und Anwendung neuronaler Netze, mit Blick auf Deep Learning. Dazu gehören unter anderem:

- Einzelne Neuronen: logistische Regression und Aktivierungsfunktionen
- Automatische Differentiation und Backpropagation
- Latente Variablen, Autoencoder
- Implementierung von Training und Inferenz mit Pytorch
- Verwenden vortrainierter Modelle

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- Grundbegriffe und -konzepte neuronaler Netze benennen und erklären,
- mathematische Grundlagen für neuronale Netze anwenden,
- einfache Modelle selbst implementieren,
- vortrainierte Modelle in andere Systeme integrieren und
- beurteilen, welche der besprochenen Modelle hinsichtlich einer Anwendung in Frage kommen.

Literatur

- Nielsen, Michael A.: Neural Networks and Deep Learning. Determination Press. San Francisco, 2015. (online)
- Goodfellow, Ian and Bengio, Yoshua and Courville, Aaron: Deep Learning. MIT. Cambridge, 2017. (online)
- Stevens, Eli and Antiga, Luca and Viehmann, Thomas: Deep Learning with PyTorch. Manning. New York, 2020 (online)
- Zhang, Aston and Lipton, Zachary C. and Li, Mu and Smola, Alexander J.: Dive into Deep Learning. Online. Everywhere, 2021. (online)

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

- Inhaltlich: Inhalte des Moduls *Data Science*

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive Mitarbeit in den Übungen
- Abgabe der Hausaufgaben
- schriftliche Klausur (in der Regel 90 Minuten) oder mündliche Prüfung am Ende des Semesters

Modulbeauftragte und hauptamtlich Lehrende

Dr. Konrad Völkel

Einführung in die Funktionale Programmierung (en. Introduction to Functional Programming)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) | jedes Wintersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

In diesem Modul werden Konzepte aus der Funktionalen Programmierung vermittelt. Dabei verwenden wir als Sprache Clojure - ein modernes Lisp, das auf der JVM läuft. Wir behandeln folgende Aspekte:

- Clojure Syntax und Programmierung
- immutable Datenstrukturen und Laziness
- das epochale Zeit-Modell
- Simplicity und die Clojure Philosophie
- Polymorphismus a la carte
- Homoikonizität und Macros.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- die Prinzipien der funktionalen Programmierung bezeichnen und klassischen, imperativen Programmiersprachen gegenüberstellen,
- bewerten, für welche Einsatzbereiche die funktionale Programmierung vorteilhaft ist,
- eigenständig funktionale Programme erstellen und testen,
- die Konzepte der unter Inhalte gelisteten Themen erklären und anwenden.

Literatur

- Moseley, Marks: Out of the tarpit. Online
- Fogus, Houser: The Joy of Clojure. Manning. 2014. 2nd edition.
- Emerick, Carper, Grand: Programming Clojure. O'Reilly. 2012. 1st edition.
- Rathore, Avila: Clojure in Action. Manning. 2016. Second 2nd.
- Higginbotham: Clojure for the Brave and True. No Starch Press. 2015. 1st edition.

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

- Inhaltlich: Inhalte der Module *Programmierung* und *Programmierpraktikum 1*

Voraussetzungen für die Vergabe von Leistungspunkten

- Je nach Teilnehmendenzahl:
- bevorzugt schriftliche Prüfung (Klausur, i. d. R. 90 Minuten),
- mündliche Prüfung (i. d. R. 30-45 Minuten).

Modulbeauftragte und hauptamtlich Lehrende

Philipp Körner, Dr. Jens Bendisposto, Prof. Dr. Michael Leuschel

Einführung in die naturwissenschaftliche Informatik (en. Introduction to Scientific Computer Science)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) | jedes Wintersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Das Modul beschreibt mit Hilfe ausgewählter Beispiele die Anwendung der Informatik und Statistik zur Lösung verschiedener Probleme aus Biologie, Physik und Chemie. Wesentliche Teile der zugrundeliegenden Algorithmen werden in der Programmiersprache Python implementiert:

- Schnelle Fourier-Transformation zur Reduzierung der Laufzeit im multiplen Alignment
- Pebble-Game-Algorithmus zur Steifigkeitsanalyse von Biomolekülen
- Motivsuche in DNA-Sequenzen mittels Gibbs-Sampling
- Dynamic-Programming im paarweisen Sequenzvergleich
- Gruppierungsverfahren für Sequenz- und Expressionsdaten: Neighbor-Joining, Markov-Clustering-Algorithmus, k-Means, Expectation Maximization
- Lateraler Gentransfer oder phylogenetische Artefakte? Statistischer Test der Übereinstimmung von Bäumen mit nicht-identischen Blattmengen ohne einen verlässlichen Referenzbaum
- Wurzeln phylogenetischer Bäume mit Hilfe des Mean-Ancestor-Deviation-Verfahrens
- Rekursion und das Problem der Independent-Phylogenetic-Contrasts

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- die zugrundeliegenden naturwissenschaftlichen Hintergründe und die damit verbundenen Problemstellungen der vorgestellten Methoden beschreiben,
- die zur Problemlösung dargestellten Algorithmen auf Beispieldaten anwenden,
- verschiedene mögliche Lösungswege zu einer Problemstellung kritisch miteinander vergleichen und
- die erlernten Verfahren eigenständig in der Programmiersprache Python implementieren.

Literatur

- Relevante Publikationen werden in der Lehrveranstaltung bekannt gegeben.

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete

Teilnahmevoraussetzungen

- keine

Voraussetzungen für die Vergabe von Leistungspunkten

- mindestens 50 Prozent der Punkte aus den Übungsaufgaben
- Bestehen der schriftlichen Abschlussprüfung (i. d. R. 90 Minuten)

Modulbeauftragte und hauptamtlich Lehrende

Dr. Mayo Röttger, Prof. Dr. Martin Lercher

Graphenalgorithmen 1 (en. Algorithms for Graphs 1)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) | jedes Wintersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Dieses Modul vermittelt grundlegende Kenntnisse aus folgenden Bereichen.

- Suchmethoden auf Graphen
- Topologische Anordnungen
- Zusammenhangsprobleme
- Kürzeste Wege
- Minimale Spannbäume
- Netzwerkfluss-Probleme
- Matching-Probleme

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- die besprochenen Graphalgorithmen wiedergeben und erklären,
- die besprochenen Graphalgorithmen unterschiedlichen Problemstellungen zuordnen und geeignet anwenden,
- die besprochenen Graphalgorithmen auf ihre Laufzeit und Korrektheit hin analysieren und
- neue einfache Graphalgorithmen entwickeln und analysieren.

Literatur

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein: Algorithmen – Eine Einführung. De Gruyter Oldenbourg. 2017. 4. Auflage.
- Ravindra Ahuja, Thomas Magnanti, James B. Orlin: Network Flows: Theory, Algorithms and Applications. Prentice Hall. Englewood Cliffs, 1993. 1. Ausgabe

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

- Inhaltlich: Inhalte der Module *Programmierung, Algorithmen und Datenstrukturen* und *Mathematik für Informatik 1*

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive Mitarbeit in den Übungen
- Abgabe der Hausaufgaben
- schriftliche Klausur (i. d. R. 90 Minuten) oder mündliche Prüfung am Ende des Semesters

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Melanie Schmidt, Dr. Daniel Schmidt

Grundlagen Verteilter Systeme (en. Foundations of Distributed Systems)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) | jedes Sommersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

- Architekturformen (Client/Server, Cloud, Fog, Edge)
- Sockets, Multithreading und Skalierbarkeit
- Entfernter Funktionsaufruf (gRPC)
- Zeit (Uhrensynchronisierung, logische Zeit, Kausalität)
- Gruppenkommunikation und Pub/Sub
- Replikation und Konsistenz (Grundlagen)
- Globale Zustände (asynchrone Schnapsschüsse, Anwendungen)
- Fehlertoleranz (Fehlererkennung und -erholung)
- Schwache Konsistenz und Skalierbarkeit (Gnutella, Chord, Dynamo, GFS, ...)
- Strenge Konsistenz und Skalierbarkeit (Transaktionen, Paxos, ...)
- Sicherheit (Grundlagen)

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- Architekturen verteilter Systeme und deren Eigenschaften erklären und vergleichen,
- verschiedene Kommunikationsmechanismen vergleichen und anwenden,
- die besprochenen verteilten Algorithmen anwenden,
- Replikations- und Konsistenz-Strategien sowie deren Skalierbarkeit erklären und vergleichen,
- Fehlermodelle und Erholungsstrategien beschreiben und
- grundlegende Sicherheitsaspekte erläutern.

Literatur

- G. Coulouris et.al., „Distributed Systems: Concepts and Design“, Addison-Wesley, 5. Aufl. 2011
- A. Tanenbaum and M. van Steen: „Distributed Systems: Principles and Paradigms“, 3. Auflage, Prentice Hall, 2013.
- Ergänzende Empfehlungen werden in der Lehrveranstaltung genannt.

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

- Inhalte des Moduls *Programmierung*

Voraussetzungen für die Vergabe von Leistungspunkten

- Bestehen der Prüfung

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Michael Schöttner

Introduction to Logic Programming (en. Introduction to Logic Programming)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|---|----------------------|------------------------------------|---------------------|
| 5 LP | 150 Stunden | 90 Stunden | 60 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) praktische Übung (2 SWS) | jedes Wintersemester | Bachelor-Studiengang Informatik | Englisch Deutsch |

Inhalte

In der imperativen Programmierung werden Programme durch Sequenzen von Instruktionen dargestellt. In der logischen Programmierung beschreibt man statt einer Lösung für ein Problem in Form eines Algorithmus das Problem durch Fakten und Regeln und der Computer findet selbständig eine Lösung. Das eröffnet einen völlig neuen und radikal unterschiedlichen Blickwinkel auf das Programmieren der auch bei der alltäglichen Entwicklung mit C oder Java sehr nützlich ist.

Die Vorlesung behandelt die Themen:

- Aussagenlogik, Prädikatenlogik
- Resolution
- Programmieren mit Horn Klauseln
- Praktische Grundlagen von Prolog
- Suchalgorithmen und Künstliche Intelligenz mit Prolog
- Grundzüge der Constraint-Programmierung

Es gibt Vorlesungsvideos auf Englisch. Die Vorlesung wird normalerweise auf Englisch gehalten. In Absprache mit allen anwesenden Studierenden kann die Vorlesung auch auf Deutsch gehalten werden.

Zusätzliche Bemerkungen

Die Folien sind auf Englisch und Deutsch verfügbar.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- die logischen Grundlagen von Prolog anwenden können um Berechnungen in Aussagenlogik und Prädikatenlogik durchzuführen
- die Datenstrukturen von Prolog zur Darstellung von Daten einsetzen können
- kleinere Prolog Programme, wie in den Übungsaufgaben besprochen, eigenständig entwickeln
- wichtige Suchalgorithmen vergleichen und praktisch in Prolog umsetzen und
- kleinere Aufgaben der symbolischen künstlichen Intelligenz in Prolog lösen.

Literatur

- Nilsson, Maluszynski: Logic, Programming and Prolog (2nd edition). (eBook)
- Blackburn, Bos, Striegnitz: Learn Prolog Now!. College Publications.

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik
- Wahlbereich im Master-Studiengang Artificial Intelligence and Data Science

Teilnahmevoraussetzungen

- Inhaltlich: Programmierkenntnisse.

Voraussetzungen für die Vergabe von Leistungspunkten

- Aktive und erfolgreiche Mitwirkung in den Übungen
- Erfolgreiche Teilnahme an der Abschlussprüfung

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Michael Leuschel

Introduction to Modelling metabolic networks (en. Introduction to Modelling metabolic networks)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) | unregelmäßig | Bachelor-Studiengang Informatik | Englisch |

Inhalte

- Introduction to the statistical programming language R
- Repetition of basic linear algebra
- Basic properties and reconstruction of stoichiometric matrices
- Topology and fundamental subspaces of the stoichiometric matrix
- Elementary flux modes
- Properties of the solution space
- Flux balance analysis
- Flux variability, flux coupling
- Modeling of gene knockouts
- Flux balance analysis with molecular crowding
- Resource balance analysis

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- summarize important constraint-based modeling techniques and apply them to metabolic networks,
- describe biological systems from possible biochemical reactions,
- formulate and solve linear optimization problems using the R programming language and
- consider metabolic modules as a system and simulate their behavior under different conditions

Literatur

- Bernhard Ø. Palsson: Systems Biology: Properties of Reconstructed Networks. Cambridge University Press. 2015.

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

- Inhaltlich: Inhalte der Module *Programmierung* und *Mathematik für Informatik 1* (oder *Lineare Algebra I*)

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive Teilnahme an den Übungen
- erfolgreiches Bearbeiten der Übungsaufgaben (50%)
- bestandene abschließende Prüfung (i. d. R. schriftlich)

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Martin Lercher

Kombinatorische Algorithmen für Clusteringprobleme (en. Combinatorial Algorithms for Clustering Problems)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) | unregelmäßig | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Die Vorlesung ist eine weiterführende Algorithmenvorlesung, die sich mit Clusteringalgorithmen beschäftigt. Wir besprechen in dieser Vorlesung vorwiegend kombinatorische Modellierungen von Clusteringproblemen und zugehörige Algorithmen. Insbesondere geht es um folgende Themen:

- Hierarchisches Clustering
- Algorithmen für das k-center Problem
- Algorithmen für das k-supplier Problem
- ähnlichkeitsbasierte Clusteringverfahren.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- kombinatorische Argumente wiedergeben und entwickeln,
- die besprochenen Clusteringalgorithmen anwenden,
- hierarchische Clusteringmethoden diskutieren und bewerten, und
- Clusteringverfahren analysieren.

Literatur

- Ausgewählte Veröffentlichungen zum Thema der Lehrveranstaltung.
- Eigenes Skript

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

- Inhaltlich: Inhalte des Moduls *Mathematik für Informatik 1*

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive Teilnahme an den Übungen
- erfolgreiches Bearbeiten der Übungsaufgaben
- abschließende Prüfung (schriftlich oder mündlich)

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Melanie Schmidt

Kryptokomplexität 1 (en. Cryptocomplexity 1)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------|------------------------------------|--------------------|
| 10 LP | 300 Stunden | 90 Stunden | 210 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (4 SWS) Übung (2 SWS) | unregelmäßig | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

In diesem Modul werden Kenntnisse über die Grundlagen der Kryptologie vermittelt, einige wichtige Kryptosysteme vorgestellt und ihre Sicherheit diskutiert, die häufig auf der Komplexität geeigneter Probleme beruht. Deshalb wird auch in die Grundlagen der Komplexitätstheorie eingeführt, insbesondere mit dem Ziel, Methoden zum Nachweis unterer Schranken bezüglich der Komplexitätsmaße Rechenzeit und Speicherplatz zu verstehen und anwenden zu können. Besonderer Wert wird darauf gelegt, dass die Studierenden die enge Verflechtung dieser Gebiete verstehen.

Einführung in die Kryptologie

- Einige klassische Kryptosysteme und ihre Kryptoanalyse (Substitutions- und Permutationschiffren, affin lineare Blockchiffren, Block- und Stromchiffren)
- Perfekte Geheimhaltung (Satz von Shannon und Vernams One-Time Pad, Entropie und Schlüsselmeerdeutigkeit)
- RSA, Primzahltests und das Faktorisierungsproblem (das Public-Key Kryptosystem RSA, digitale Signaturen mit RSA, Sicherheit von RSA)

Einführung in die Komplexitätstheorie

- Grundlagen (Komplexitätsmaße und -klassen, Kompression und Beschleunigung, Hierarchiesätze)
- Zwischen L und PSPACE (einfache Inklusionen, komplexitätsbeschränkte Many-one-Reduktionen, vollständige Probleme in NL, NP-vollständige Probleme)

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- symmetrische Kryptosysteme hinsichtlich ihrer Eigenschaften klassifizieren,
- die Sicherheit von Blockchiffren und anderen klassischen Kryptosystemen bewerten,
- begründen, weshalb bestimmte Kryptosysteme welche Eigenschaften besitzen,
- die Idee der Public-Key-Kryptographie erläutern,
- die grundlegenden Ziele und Definitionen der Komplexitätstheorie erläutern,
- die Komplexität von natürlichen Problemen beschreiben und begründen und
- selbstständig Reduktionen zwischen Problemen zum Nachweis unterer Schranken entwerfen und ihre Korrektheit nachweisen.

Literatur

- Jörg Rothe: Komplexitätstheorie und Kryptologie. Eine Einführung in Kryptokomplexität. eXamen.Press. Springer-Verlag, Heidelberg, 2008. 1. Auflage.
- Jörg Rothe: Complexity Theory and Cryptology. An Introduction to Crypto-complexity. EATCS Texts in Theoretical Computer Science, Springer-Verlag. Heidelberg, 2005. 1. Auflage.
- Ergänzende Empfehlungen werden in der Lehrveranstaltung genannt.

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

- keine

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive und erfolgreiche Mitwirkung in den Übungen
- schriftliche Prüfung (Klausur, i. d. R. 90 Minuten)

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Jörg Rothe

Muster in der Natur: theoretische Hintergründe und Algorithmen (en. Patterns in nature: theoretical background and algorithms)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) | unregelmäßig | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

In der Lehrveranstaltung werden in der Natur vorkommende Muster betrachtet, im Hinblick auf ihre Entstehung, die zugrundeliegenden chemischen/physikalischen Prozesse, mögliche mathematische Modellierung und die Algorithmen um diese am Computer nachzubilden.

- natürliche Muster (Symmetrie, Fraktale, Spiralen, Chaos, Parkettierung, Punkte und Streifen (Turing Patterns))
- die zugrundeliegenden chemischen/physikalischen Prozesse, mathematische Modellierung (z. B. Reaktionsdiffusionsgleichung, Fibonacci Zahlen und goldener Schnitt)
- Algorithmen um diese am Computer nachzubilden (wie z. B. zelluläre Automaten, L-Systeme, Chaos Game).

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden

- die Verschieden in der Natur vorkommenden Muster erkennen, und erklären wie diese entstehen
- die gängigen Algorithmen zur Erzeugung dieser Muster implementieren und einsetzen

Literatur

- Ergänzende Empfehlungen werden in der Lehrveranstaltung genannt.
- Eigenes Skript

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

- Inhaltlich: Inhalte der Module *Programmierung, Algorithmen und Datenstrukturen* und *Mathematik für Informatik 1* (oder *Lineare Algebra I* oder *Analysis I*)

Voraussetzungen für die Vergabe von Leistungspunkten

- Gruppenarbeit mit Präsentation; zu 20% gewichtet

- Ausarbeitung zur Gruppenarbeit am Ende des Semester; zu 80% gewichtet

Modulbeauftragte und hauptamtlich Lehrende

Dr. Hannah Franziska Löchel

Präferenzaggregation durch Wählen: Algorithmik und Komplexität (en. Preference Aggregation by Voting: Algorithmics and Complexity)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------|------------------------------------|--------------------|
| 10 LP | 300 Stunden | 90 Stunden | 210 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (4 SWS) Übung (2 SWS) | unregelmäßig | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

In diesem Modul werden die wichtigsten Grundlagen, Modelle, Methoden und Resultate aus dem jungen, sich rasant entwickelnden, interdisziplinären Gebiet *Computational Social Choice* vermittelt, das Anwendungen in den Gebieten *Artificial Intelligence* und *Multiagent Systems* hat. Insbesondere werden Eigenschaften von Wahlsystemen behandelt sowie die damit im Zusammenhang stehenden Entscheidungsprobleme (Gewinnerbestimmung, Manipulation, Wahlkontrolle, Bestechung usw.) hinsichtlich ihrer Algorithmik und Komplexität untersucht.

Grundlagen der Social-Choice-Theorie

- Wahlen
- Wahlsysteme (Scoring-Protokolle wie Plurality, Veto und Borda; Condorcet; Copeland; Maximin; Dodgson; Young; Bucklin; Fallback; etc.)
- Eigenschaften von Wahlsystemen
- Einige Wahl-Paradoxa und Unmöglichkeitssätze

Algorithmik und Komplexität von Wahlproblemen

- Gewinnerbestimmung
- Mögliche und notwendige Gewinner
- Manipulation
- Wahlkontrolle
- Bestechung

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- die gebräuchlichsten Wahlsysteme (Scoring-Protokolle wie Plurality, Veto und Borda; Condorcet; Copeland; Maximin; Dodgson; Young; Bucklin; Fallback; etc.) und ihre Eigenschaften beschreiben,
- diskutieren, wie sinnvoll welche axiomatischen Eigenschaften von Wahlsystemen sind,
- die Gewinner der gebräuchlichsten Wahlsysteme für gegebene Präferenzprofile bestimmen,
- Beispiele für erfolgreiche Manipulations-, Kontroll- und Bestechungsaktionen für die gebräuchlichsten Wahlsysteme angeben und die zugehörigen Entscheidungsprobleme beschreiben,
- argumentieren, weshalb welche dieser Probleme entweder durch effiziente Algorithmen gelöst werden können oder schwer zu lösen sind.

Literatur

- Jörg Rothe (ed.): *Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*. Springer Texts in Business and Economics, Springer-Verlag. Heidelberg, 2015. 1. Auflage.

- Jörg Rothe, Dorothea Baumeister, Claudia Lindner und Irene Rothe: Einführung in Computational Social Choice. Individuelle Strategien und kollektive Entscheidungen beim Spielen, Wählen und Teilen. Spektrum Akademischer Verlag (Springer). Heidelberg, 2011. 1. Auflage.
- Ergänzende Empfehlungen werden in der Lehrveranstaltung genannt.

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

- keine

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive und erfolgreiche Mitwirkung in den Übungen
- schriftliche Prüfung (Klausur, i. d. R. 90 Minuten)

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Jörg Rothe

Randomisierte Algorithmen und Analysetechniken (en. Randomized Algorithms and Analysis)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------|------------------------------------|--------------------|
| 10 LP | 300 Stunden | 90 Stunden | 210 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (4 SWS) Übung (2 SWS) | unregelmäßig | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Die Veranstaltung ist eine weiterführende Algorithmenvorlesung, die sich mit Randomisierung als Entwurfs- und Analyseverfahren beschäftigt. Ziel ist es, durch zufällige Entscheidungen effiziente Algorithmen zu erhalten, die deutlich schneller als deterministische Varianten arbeiten und dabei mit möglichst hoher Wahrscheinlichkeit genaue Ergebnisse ausgeben.

- randomisierte Algorithmenmodelle (Las-Vegas- und Monte-Carlo-Algorithmen)
- Laufzeit und Genauigkeitsanalyse
- randomisierte Approximationsalgorithmen (z. B. für SAT- und Graphprobleme)
- Methoden zur Wahrscheinlichkeitsverstärkung
- randomisierte Entwurfsmethoden (z. B. probabilistische Methode, Fingerabdrücke, Hashing)
- Randomisierung in der Datenanalyse

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden: - Begriffe und grundlegende Methoden der Randomisierung in der Algorithmik beschreiben, - Algorithmen unterschiedlichen Entwurfparadigmen zuordnen und exemplarisch anwenden, - Eigenschaften wie Laufzeiten und Genauigkeiten randomisierter Algorithmen analysieren und dadurch bewerten und vergleichen und - erste randomisierte Algorithmen nach geeigneten Entwurfsmustern entwickeln.

Literatur

- Juraj Hromkovič: Randomisierte Algorithmen: Methoden zum Entwurf von zufallsgesteuerten Systemen für Einsteiger. Teubner-Verlag. Wiesbaden, 2004. 1. Ausgabe
- Michael Mitzenmacher and Eli Upfal: Probability and Computing. Cambridge University Press. Cambridge, 2017. 2nd Edition.

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

- Inhaltlich: Inhalte der Module *Algorithmen und Datenstrukturen*, *Theoretische Informatik* und *Mathematik für Informatik 3*

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive Teilnahme an den Übungen
- erfolgreiches Bearbeiten der Übungsaufgaben
- abschließende Prüfung (schriftlich oder mündlich)

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Melanie Schmidt, Dr. Anja Rey

Statistische Datenanalyse (en. Statistical Data Analysis)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) | jedes Sommersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Das Modul baut auf einem Kurs und dem zugehörigen Buch von Prof. Gianluca Bon-tempi an der Université Libre de Bruxelles zu den statistischen Grundlagen des maschinellen Lernens auf. Inhalte sind im Einzelnen:

- Die Programmiersprache R für Statistical Computing
- Beschreibende Statistik
- Grundlagen der Wahrscheinlichkeitsrechnung
- Klassische parametrische Schätzung und Tests
- Nicht-parametrische Schätzung und Tests
- Statistisches Lernen
- Lineare Ansätze
- Nicht-lineare Ansätze
- Dimensionalitätsreduktion

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- grundlegende Konzepte der statistischen Datenanalyse zusammenfassen (z. B. Wahrscheinlichkeitsverteilungen, parametrische und nichtparametrische Hypothesentests, Parameterschätzer, bedingte Wahrscheinlichkeiten, Permutationstests),
- parametrische und nicht-parametrische Tests für statistische Hypothesen entwickeln,
- statistische Modelle mit mehrdimensionalen Prädiktoren zu formulieren,
- statistische Datenanalysen mit R planen und durchführen und
- aussagekräftige grafische Darstellungen von Daten mit ggplot2 erstellen.

Literatur

- Gianluca Bontempi. Handbook statistical foundations of machine learning. Self-published, Brüssel, 2021. 2nd edition.

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

- keine

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive Teilnahme an den Übungen
- erfolgreiches Bearbeiten der Übungsaufgaben (50%)
- abschließende Prüfung (i. d. R. schriftlich)

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Martin Lercher

Von der Schaltung zur Software (en. From Circuits to Software)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|-------------------------|------------------------------------|--------------------|
| 10 LP | 300 Stunden | 90 Stunden | 210 Stunden |
| Lehrveranstaltungen | Häufigkeit des Angebots | Studiengang | Unterrichtssprache |
| Vorlesung (4 SWS) Übung (2 SWS) | jedes Sommersemester | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Diese Veranstaltung bietet eine Reise durch verschiedene Gebiete der Informatik, um dabei ein Gesamtbild über die prinzipielle Funktionsweise von Computern vermitteln. Ausgehend von dem NAND-Gatter werden alle wichtigen Hardware-Komponenten eines Computers in einer Simulation nachgebaut, ALU, CPU, RAM, etc. Für die im Rahmen der Vorlesung entwickelte CPU und die dazugehörige Maschinensprache werden schrittweise eine Assemblersprache, eine virtuelle Maschine und eine Programmiersprache sowie der dazugehörige Compiler entwickelt. Mit all diesen Werkzeugen werden schließlich ein einfaches Betriebssystem und Anwendungsprogramme entwickelt.

Die Vorlesung wird begleitet von Übungen. Die eigenständige praktische Anwendung des Gelernten soll im Fokus der Veranstaltung liegen. Die Studierenden entwickeln in den Übungen die diversen in der Vorlesung vorgestellten Komponenten moderner Computer.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- die Grundprinzipien aller besprochenen Ebenen erläutern und bewerten und
- eigenständig neue Funktionalitäten auf diesen Ebenen entwickeln.

Literatur

- Noam Nisan, Shimon Schocken: The Elements of Computing Systems: Building a Modern Computer from First Principles. MIT Press. Cambridge, 2008. 1st Edition

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016

Teilnahmevoraussetzungen

- Inhaltlich: Inhalte des Moduls *Programmierung*

Voraussetzungen für die Vergabe von Leistungspunkten

- erfolgreiches Bearbeiten der Übungsaufgaben
- abschließende Prüfung (i. d. R. schriftlich)

Modulbeauftragte und hauptamtlich Lehrende

Dr. John Witulski

Seminar (en. Seminar)

Das Seminar dient der Vorbereitung auf die Bachelorarbeit und sollte demnach vor Anmeldung dieser abgeschlossen sein.

Diese Veranstaltungen sind hinsichtlich Belegung und Anwesenheit anders als die meisten anderen Veranstaltungen des Instituts für Informatik. Dies bedeutet konkret:

- Pro Seminargruppe sind die Plätze auf 25 Teilnehmer*innen beschränkt.
- Diese Plätze werden in der Regel vor Vorlesungsbeginn vergeben und die Platzvergabe ist mit Vorlesungsbeginn abgeschlossen.
- Die genauen Belegungsfristen können Sie HIS/LSF entnehmen.
- In Seminaren herrscht, entsprechend der Prüfungsordnung, Anwesenheitspflicht.

Bachelor-Seminar (en. Bachelor's-Seminar)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|---|----------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Seminar (2 SWS) Seminar / Vorlesung / Übung (2 SWS) | jedes Semester | Bachelor-Studiengang Informatik | Deutsch, Englisch |

Inhalte

In einer Lehrveranstaltung des Moduls „Seminar“ eignen sich Studierende selbstständig mithilfe von Literatur und anderen Quellen Wissen und Methoden zu einem Thema der Informatik an. Sie üben die Präsentation der zuvor unbekannt und nun selbstständig erlernten Inhalte in Vortragsform, erlernen das Verfassen von wissenschaftlichen Texten und geben anderen Studierenden konstruktives Feedback. Sowohl die mündliche, als auch die schriftliche Präsentation des selbstständig erarbeiteten Themas dienen der Vorbereitung auf die Bachelorarbeit.

Seminare werden von allen Arbeitsgruppen der Informatik zu verschiedenen Themen angeboten. Die konkreten, thematischen Inhalte eines Seminars können den in HIS/LSF verlinkten Veranstaltungsbeschreibungen entnommen werden.

Die Vortrags- sowie Ausarbeitungsleistungen fließen in die Seminarnote ein. Die Bewertungskriterien werden zu Beginn des Seminars kommuniziert.

Zusätzliche Bemerkungen

- Die Unterrichtssprache des Kurses wird in den in HIS/LSF verlinkten Informationen angegeben.
- Weitere Informationen zu den Themen des Seminars unter <https://www.cs.hhu.de/bachelor21/seminar>.
- Die Gewichtung der Kriterien für die Vergabe von Leistungspunkten für die Note wird in der ersten oder zweiten Sitzung bekannt gegeben.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- selbstständig relevante, qualitativ angemessene Literatur, Methoden und Quellen zu einem vorgegebenen Fachthema finden, auswählen, auswerten, zusammenfassen und sich damit kritisch auseinandersetzen
- dieses Thema zielgruppengerecht in einem Vortrag unter Einhaltung einer zeitlichen Vorgabe präsentieren
- anderen Personen konstruktiv Feedback zu einem Fachvortrag geben
- mit anderen Informatiker:innen kritisch ein Fachthema diskutieren
- ein Fachthema in einem kohärenten, strukturierten Fachtext unter Einhaltung formaler Kriterien erläutern, einordnen und diskutieren

Literatur

- J. Zobel: Writing for Computer Science. Springer. 2014. 3. Auflage
- Ausgewählte Veröffentlichungen zum Thema der Lehrveranstaltung.

Verwendbarkeit des Moduls

- Bereich Seminar Bachelor-Studiengang PO 2021

Teilnahmevoraussetzungen

- Formal: Erfolgreicher Abschluss des Moduls *Wissenschaftliches Arbeiten*
- Inhaltlich: siehe im HIS/LSF verlinkte Seminarinformationen

Voraussetzungen für die Vergabe von Leistungspunkten

- mind. ausreichende schriftliche Ausarbeitung
- mind. ausreichender Vortrag
- aktive Teilnahme an den Seminar-Terminen
- Anwesenheit bei 80% der Seminarvorträge

Modulbeauftragte und hauptamtlich Lehrende

Lehrende der Informatik

Bachelorarbeit (en. Bachelor Thesis)

Das Modul Bachelorarbeit besteht aus zwei Teilbereichen, die beide benotet werden.

Abschlussarbeit (en. Thesis)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|---------------------|----------------|------------------------------------|---------------------|
| 12 LP | 360 Stunden | - | - |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| - | jederzeit | Bachelor-Studiengang Informatik | Deutsch Englisch |

Inhalte

Der Inhalt der Bachelor-Arbeit liegt in einem gewählten Modul der Informatik. Die Bachelor-Arbeit muss in deutscher oder englischer Sprache verfasst werden.

Lernergebnisse/Kompetenzen

Mit der schriftlichen Abschlussarbeit sollen die Studierenden nachweisen, dass sie in der Lage sind:

- innerhalb einer vorgegebenen Frist (von 3 Monaten)
- ein Thema selbstständig zu bearbeiten und
- dieses angemessen darzustellen.

Literatur

- In Absprache mit den Betreuungspersonen.

Verwendbarkeit des Moduls

- Bachelorarbeit

Teilnahmevoraussetzungen

Für die Anmeldung zur Bachelor-Arbeit müssen mindestens 120 der im Rahmen des Bachelor-Studiums zu erwerbenden Leistungspunkte nachgewiesen werden.

Voraussetzungen für die Vergabe von Leistungspunkten

- erfolgreiche Bearbeitung des Themas und Darstellung in einer fristgerecht abgegebenen Ausarbeitung (Bachelor-Arbeit)
- Präsentation in einem mündlichen Vortrag mit Diskussion

Modulbeauftragte und hauptamtlich Lehrende

Lehrende der Informatik

Abschlussseminar (en. Final Seminar)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|---------------------|----------------|------------------------------------|---------------------|
| 3 LP | 90 Stunden | 30 | 60 |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| - | jederzeit | Bachelor-Studiengang Informatik | Deutsch Englisch |

Inhalte

Im Rahmen des benoteten Abschlussseminars werden Thema und Ergebnisse der Abschlussarbeit der entsprechenden Arbeitsgruppe (oder einem Zusammenschluss solcher) mit anschließender wissenschaftlicher Diskussion präsentiert.

Lernergebnisse/Kompetenzen

Nach der Teilnahme am Abschlussseminar sind die Studierenden in der Lage

- ein selbstständig erarbeitetes Thema in einem Vortrag zu präsentieren
- wissenschaftlichen Vorträgen anderer Absolvent*innen zu folgen
- sich in die Diskussion zu wissenschaftlichen Vorträgen anderer Absolvent*innen einzubringen

Literatur

-

Verwendbarkeit des Moduls

- Bachelorarbeit

Teilnahmevoraussetzungen

Die Abschlussarbeit muss angemeldet sein.

Voraussetzungen für die Vergabe von Leistungspunkten

Aktive Teilnahme am Abschlussseminar und Präsentation des eigenen Themas in einem mündlichen Vortrag mit Diskussion.

Modulbeauftragte und hauptamtlich Lehrende

Lehrende der Informatik

Nicht mehr angebotene Module

In diesem Kapitel finden Sie alle Module, die wir nicht mehr anbieten.

Bachelor-Seminar: Einführung in die Blockchaintechnologie (en. Bachelor's Seminar: Introduction to blockchain technology)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|----------------------------------|----------------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Übung (2 SWS) Seminar (2 SWS) | nicht mehr angeboten | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

- Regeln für konstruktives Feedback
- Grundlagen der Blockchaintechnologie (Hashfunktionen, Signaturen, Proof of Work, ...)
- dezentrale Währungen (z. B. Bitcoin)
- weitere Inhalte werden in Absprache mit den Studierenden anhand der vergebenen Seminarvorträge festgelegt (z. B. Angriffe, Abwehrstrategien, verschiedene Protokolle)

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- sich selbstständig in ein vorgegebenes Themengebiet einarbeiten,
- dieses Thema schriftlich darstellen und in einem Vortrag vorstellen,
- Kritik zur eigenen Ausarbeitung in eine Überarbeitung dieser einfließen lassen,
- konstruktives Feedback zu anderen Ausarbeitungen und Vorträgen geben,
- die Grundlagen der Blockchain Technologie erläutern,
- verschiedene Anwendungsgebiete der Blockchain nennen und begründen, und
- die Inhalte der einzelnen Seminarvorträge zusammenfassen.

Literatur

- Ausgewählte Veröffentlichungen zum Thema der Lehrveranstaltung.

Verwendbarkeit des Moduls

- Seminar Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016

Teilnahmevoraussetzungen

- Formal: Erfolgreicher Abschluss des Moduls Wissenschaftliches Arbeiten

Voraussetzungen für die Vergabe von Leistungspunkten

- min. ausreichende Seminarausarbeitung (diese wird allen Teilnehmer*innen des Kurses zur Verfügung gestellt)
- Teilnahme am Peer Review Prozess der Ausarbeitungen
- min. ausreichender Seminarvortrag
- bestehen einer mündliche Prüfung

Modulbeauftragte und hauptamtlich Lehrende

Janine Golov

Bachelor-Seminar: Programmiersprachen (en. Bachelor's-Seminar: Programming Languages)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|----------------------------------|----------------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Seminar (2 SWS) Übung (2 SWS) | nicht mehr angeboten | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Das Bachelor-Seminar Programmiersprachen vermittelt Allgemeinwissen über Programmiersprachen sowie über deren typische Eigenschaften und Konzepte. Das Seminar besteht hierbei einerseits aus einer Reihe von Vorträgen in denen Teilnehmende eine Sprache vorstellen und andererseits einer Übung in der diese Sprache durch Programmieraufgaben angewendet wird. Das Verfassen einer Ausarbeitung zu Vortragsprache bereitet Teilnehmende auf die Bachelorarbeit vor.

- Allgemeinwissen zu Programmiersprachen (Eigenschaften, Unterschiede, Anwendungen/Einsatzgebiete, Syntax und Semantik)
- Eigenschaften und Konzepte von Programmiersprachen
- einen Vortrag über ein selbständig erarbeitetes Thema halten, sowie eine Ausarbeitung zu diesem verfassen
- Verfassen einer schriftlichen Ausarbeitung zu einer Programmiersprache

Zusätzliche Bemerkungen

- Studierende, die nicht nach PO 2021 studieren, werden bei der Vergabe der Plätze mit niedrigerer Priorität berücksichtigt.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- Eigenschaften der besprochenen Programmiersprachen beschreiben und erklären,
- selbstständig einen Vortrag und eine Ausarbeitung zu einem gegebenen Thema erstellen und
- diesen Vortrag halten.

Literatur

- Bruce A. Tate. 2010: Seven Languages in Seven Weeks: A Pragmatic Guide to Learning Programming Languages. Pragmatic Bookshelf. 1st edition

Verwendbarkeit des Moduls

- Bereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016

Teilnahmevoraussetzungen

- Inhaltlich: Inhalte der Module Programmierung

Voraussetzungen für die Vergabe von Leistungspunkten

- Erstellen eine Ausarbeitung
- Halten eines Vortrags
- Bearbeitung von Programmieraufgaben

Modulbeauftragte und hauptamtlich Lehrende

Dr. John Witulski

Bachelor-Seminar: Überblick Künstliche Intelligenz (en. Bachelor's-Seminar: Introduction to Artificial Intelligence)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|--|----------------------|---|---------------------------|
| 5 LP | 150 Stunden | 90 Stunden | 60 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung/Seminar (2 SWS) Übung (2 SWS) | nicht mehr angeboten | Bachelor-Studiengang Informatik und Master DSAI | Deutsch / Englisch (DSAI) |

Inhalte

Mögliche Inhalte des Seminars, die in Absprache mit den Teilnehmern behandelt werden können sind:

- Regeln für konstruktives Feedback
- Grundlagen der künstlichen Intelligenz, von klassischer symbolischer KI bis hin zu modernen Techniken wie Deep Learning
- weitere Inhalte werden in Absprache mit den Studierenden anhand der vergebenen Seminarvorträge festgelegt

Zusätzliche Bemerkungen

- Studierende, die nicht Informatik nach PO 2021 studieren, werden bei der Vergabe der Plätze mit niedrigerer Priorität berücksichtigt.

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- sich selbstständig in ein vorgegebenes Themengebiet einarbeiten,
- dieses Thema schriftlich darstellen und in einem Vortrag vorstellen,
- Kritik zur eigenen Ausarbeitung in eine Überarbeitung dieser einfließen lassen,
- konstruktives Feedback zu anderen Ausarbeitungen und Vorträgen geben,
- Fähigkeiten und problemspezifische Anwendbarkeit von wichtigen KI Ansätzen einschätzen können (z. B. Expertensysteme, SVMs, Decision Trees, Random Forests, CNNs, ...),
- die Zielsetzungen und Kerntechniken von Teilgebiete der KI (von klassischer symbolischer KI bis hin zu Deep Learning) erklären können,
- Probleme und Grenzen aktuelle KI Ansätze nennen und erklären (Bias, Erklärbarkeit, Sicherheit, ...) und deren Relevanz für konkrete praktische Fragen einschätzen können.

Literatur

- Bratko: Prolog Programming for Artificial Intelligence. Addison Wesley
- Russel, Norvig: Artificial Intelligence: A Modern Approach. Prentice Hall
- Szeliski: Computer Vision: Algorithms and Applications. Springer
- Goodfellow, Bengio, Courville: Deep Learning. MIT Press

Verwendbarkeit des Moduls

- Bereich Seminar Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete

- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik
- Wahlbereich im Master-Studiengang Artificial Intelligence and Data Science

Teilnahmevoraussetzungen

- Formal: Erfolgreicher Abschluss des Moduls Wissenschaftliches Arbeiten

Voraussetzungen für die Vergabe von Leistungspunkten

- Seminarvortrag
- erfolgreiche Erstellung einer Ausarbeitung
- Bestehen der Klausur

Modulbeauftragte und hauptamtlich Lehrende

Prof. Dr. Michael Leuschel

Kollektive Entscheidungen (en. Collective Decisions)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) | nicht mehr angeboten | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Dieses Modul beschäftigt sich mit verschiedenen Verfahren zur kollektiven Entscheidungsfindung unter Berücksichtigung von Präferenzen. Solche Verfahren werden in vielen Bereichen der künstlichen Intelligenz, zum Beispiel bei der Interaktion von autonomen Agenten benötigt. Inhalt der Veranstaltung sind verschiedene Verfahren mit ihren axiomatischen und algorithmischen Eigenschaften aus den drei Bereichen: Wahlen, Bürgerhaushalte und Verteilung von Gütern.

- Wahlen (Wahlregeln, Eigenschaften, Unmöglichkeitsresultate)
- Bürgerhaushalte (Präferenzen, Aggregationsregeln, Eigenschaften)
- Verteilung von Gütern (Präferenzen, soziale Wohlfahrt, Allokationen)

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- unterschiedliche Verfahren zur kollektiven Entscheidungsfindung beschreiben und auf konkrete Situationen anwenden,
- eigene Entscheidungsverfahren entwerfen und im Hinblick auf ihre Eigenschaften analysieren und
- unter gegebenen Rahmenbedingungen begründete Empfehlungen für den Einsatz bestimmter Verfahren geben.

Literatur

- Jörg Rothe, Dorothea Baumeister, Claudia Lindner und Irene Rothe: Einführung in Computational Social Choice. Individuelle Strategien und kollektive Entscheidungen beim Spielen, Wählen und Teilen. Spektrum, Akademischer Verlag. 2011.
- Felix Brandt, Vincent Conitzer, Ulle Endriss, Jerome Lang, and Ariel Procaccia (eds.): Handbook of Computational Social Choice. Cambridge University Press. 2015.

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

- Inhaltlich: Inhalte des Moduls *Theoretische Informatik*

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive und erfolgreiche Mitwirkung in den theoretischen Übungen
- schriftliche Prüfung (Klausur, i. d. R. 90 Minuten)

Modulbeauftragte und hauptamtlich Lehrende

apl. Prof. Dr. Dorothea Baumeister

Matching (en. Matching)

| Leistungspunkte | Arbeitsaufwand | Kontaktzeit | Selbststudium |
|------------------------------------|----------------------|------------------------------------|--------------------|
| 5 LP | 150 Stunden | 60 Stunden | 90 Stunden |
| Lehrveranstaltungen | Turnus | Studiengang | Unterrichtssprache |
| Vorlesung (2 SWS) Übung (2 SWS) | nicht mehr angeboten | Bachelor-Studiengang Informatik | Deutsch |

Inhalte

Dieses Modul beschäftigt sich mit unterschiedlichen Arten von Matchings bzw. Zuteilungsverfahren. Solche Verfahren können in unterschiedlichen Bereichen eingesetzt werden, Beispiele sind die Vergabe von Studienplätzen, die Verteilung von Assistenzärzten auf Krankenhäuser oder die Zuordnung von Wohnheimplätzen. Inhalt der Veranstaltung sind die verschiedenen Verfahren mit ihren axiomatischen und algorithmischen Eigenschaften.

- Matching in Graphen
- Matching mit beidseitigen Präferenzen (stabiles Heiratsproblem)
- Nicht-bipartites Matching mit Präferenzen (stabiles Mitbewohnerproblem)
- Bipartites Matching mit einseitigen Präferenzen (Vergabe von Plätzen im Studierendenwohnheim)

Lernergebnisse/Kompetenzen

Nach erfolgreicher Teilnahme an den Veranstaltungen dieses Moduls können die Studierenden:

- Zuteilungsverfahren in unterschiedlichen Situationen durchführen,
- Herausforderungen von praktischen Zuteilungsproblemen identifizieren,
- Verfahren zur Zuteilung für spezielle Einsatzbereiche entwickeln und im Hinblick auf ihre Eigenschaften untersuchen,
- bekannte Verfahren zur Zuteilung auf neue Einsatzbereiche übertragen,
- verschiedenen Verfahren zur Zuteilung gegenüberstellen und
- Empfehlungen für bestimmte Einsatzbereiche der Zuteilung geben.

Literatur

- Dan Gusfield and Robert W. Irving: The Stable Marriage Problem, Structure and Algorithms. MIT Press. 1989.
- David F. Manlove: Algorithmics of Matching under Preferences. World Scientific Publishing Company. 2013.

Verwendbarkeit des Moduls

- Wahlbereich Bachelor-Studiengang PO 2021
- Wahlpflicht- und Schwerpunktbereich Bachelor-Studiengang PO 2013 und PO 2016
- Anwendungsfach im Bachelor-Studiengang Mathematik und Anwendungsgebiete
- Nebenfach im Bachelor-Studiengang Physik
- Nebenfach im Bachelor-Studiengang Medizinische Physik

Teilnahmevoraussetzungen

- Inhaltlich: Inhalte des Moduls *Theoretische Informatik*

Voraussetzungen für die Vergabe von Leistungspunkten

- aktive und erfolgreiche Mitwirkung in den theoretischen Übungen
- schriftliche Prüfung (Klausur, i. d. R. 90 Minuten)

Modulbeauftragte und hauptamtlich Lehrende

apl. Prof. Dr. Dorothea Baumeister