

Modulhandbuch

»Informatik«

Bachelor

SPO IN2025



Entwurf - Voraussichtlicher Veröffentlichungstermin: Oktober 2025

Die Modulbeschreibungen dienen der inhaltlichen Orientierung in Ihrem Studium.

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

Inhaltsverzeichnis

1	Orientierungsphase	3
1.1	Einführung in die Programmierung	3
1.2	Entwicklungslabor	6
1.3	Mathematik 1	8
1.4	Einführung in die Theoretische Informatik	10
1.5	Rechnerstrukturen	14
1.6	Objektorientierte Programmierung	18
1.7	Software Engineering	22
1.8	Mathematik 2	26
1.9	Algorithmen und Datenstrukturen	28
1.10	Rechnerarchitektur	32
2	Vertiefungsphase	34
2.1	Fortgeschrittene Programmiertechniken	34
2.2	Softwaremodellierung	38
2.3	Statistik	40
2.4	Daten- & Informationsanalyse	44
2.5	Computernetzwerke	48
2.6	Datenbanksysteme	50
2.7	Angewandtes Software Engineering	54
2.8	Sichere Softwareentwicklung	58
2.9	Projektarbeit 1	60
2.10	Einführung in das maschinelle Lernen	62
2.11	Projektarbeit 2	66
2.12	Betriebssysteme	68
2.13	Seminar Neue Technologien	72
2.14	Fachwissenschaftliche Wahlpflichtfächer	74
2.15	Allgemeinwissenschaftliches Wahlpflichtfach	76
2.16	Praktische Tätigkeit - Praxissemester	78
2.17	Praxisseminar	80
2.18	Bachelorarbeit	82
2.19	Bachelorseminar	84

1 Orientierungsphase

1.1 Einführung in die Programmierung

Name / engl.

Einführung in die Programmierung / *Introduction to Programming*

Kürzel

1.1

Verantwortlicher

Prof. Dr. Lothar Braun

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Wintersemester

Arbeitsaufwand / Zusammensetzung

SWS: 6, CPs: 10,

Präsenzzeit: 90 h, Selbststudium: 210 h, Gesamtaufwand: 300 h

Lehrveranstaltungen

Einführung in die Programmierung (4 SWS)

Praktikum Programmieren (2 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Praktikum

Prüfung

Prüfungsnummer

-

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Portfolioprüfung:

1. Elektronische Prüfung, 60 Minuten, keine Hilfsmittel
 2. Praktische Prüfung, Arbeitsumfang: 22 h
-

Ergänzende Hinweise zur Prüfungsform

Die Note für das Modul wird wie folgt aus den beiden Teilleistungen gebildet:

- Wird der zweite Teil nicht bestanden, gilt das Modul als nicht bestanden.
 - Wird der zweite Teil bestanden, wird als Modulnote die Note des ersten Teils vergeben.
-

Inhalte des Moduls

Systematische Einführung in Syntax, Semantik und Pragmatik einer Programmiersprache (Java). Es werden die grundlegenden

Die wichtigsten Themenbereiche:

- Entwicklungswerkzeuge: Compiler, Interpreter, Debugger, IDE
- Primitive Datentypen und Variablen
- Operatoren
- Kontrollstrukturen
- Felder und Zeichenketten
- Funktionen und Rekursion
- Klassen, Objekte und Methoden
- Attribute und Kapselung
- Einfache Datenstrukturen

Die in der Vorlesung präsentierten Inhalte werden in einem begleitenden Praktikum an diversen Aufgaben eingeübt und vertieft.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Nach erfolgreichem Abschluss des Moduls sind die Studierenden in der Lage:

- Schlüsselwörter der Programmiersprache Java und deren Funktion zu beschreiben.
- Quellcode niedriger bis mittlerer Komplexität in Java zu verstehen.
- vorgegebene Algorithmen in Java selbständig und effizient zu implementieren.
- einfache Algorithmen eigenständig zu entwerfen und umzusetzen.
- Fehler in ihren Programmen zu identifizieren und zu beheben.

Literaturliste

Goll, J. et al.: *Java als erste Programmiersprache*. Teubner Verlag, Stuttgart, 2016. DOI: <https://doi.org/10.1007/978-3-658-12118-1>

Ullenboom, Christian: *Java ist auch eine Insel: Das umfassende Handbuch*. Galileo Computing, Bonn, 2019. Online-Ausgabe verfügbar unter: <http://www.tutego.com/javabuch/online.htm>.

1.2 Entwicklungslabor

Name / engl.

Entwicklungslabor / Development Lab

Kürzel

1.2

Verantwortlicher

Prof. Dr. Rolf Winter

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Wintersemester

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 5,

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

Lehrveranstaltungen

Entwicklungslabor (2 SWS)

Praktikum Entwicklungslabor (2 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Praktikum

Prüfung

Prüfungsnummer

-

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Praktische Prüfung (Umfang: 22 Stunden)

Inhalte des Moduls

Einführung in die Funktionsweise und Handhabung elementarer Werkzeuge der Informatik, die fachübergreifend zum Einsatz kommen.

Dabei ist ein besondere Schwerpunkt auf folgende Werkzeuge gelegt:

- Die (Linux-)Shell und ihre Kommandos
- Die verteilte Versionsverwaltung git
- Remote Shell-Zugriff mit ssh
- Text editieren mit vim
- Regular expressions
- Build-Management mit make
- Containervirtualisierung mit Docker

Zusätzlich werden eine Anzahl kleinerer Werkzeuge und gängiger Kniffe eingeführt und demonstriert. In einem begleitenden Praktikum werden die erlernten Werkzeuge und Techniken mit Hilfe von diversen Aufgaben eingeübt und das Verständnis dieser vertieft.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Nach Abschluss des Moduls sind Studierende in der Lage:

- die Basisfunktionen der wichtigsten Werkzeuge in der Informatik sicher anzuwenden.
- auch ohne das Internet zu bemühen deren Hilfesysteme zu nutzen, um sich auch komplexere Anwendungsfälle zu erschliessen.

Literaturliste

Michael Kerrisk: The Linux Programming Interface: A Linux and UNIX System Programming Handbook, No Starch Press, 2010

The Open Group Base Specifications Issue 8, IEEE Std 1003.1-2024 Edition,
<https://pubs.opengroup.org/onlinepubs/9799919799/>

Scott Chacon and Ben Straub: Pro Git, Apress, 2nd edition, <https://git-scm.com/book/en/v2>

Arnold Robbins and Elbert Hannah: Learning the vi and Vim Editors: Power and Agility Beyond Just Text Editing, O'Reilly Media, 28. Dezember 2021

GNU make, https://www.gnu.org/software/make/manual/html_node/index.html

Sean P. Kane and Karl Matthias: Docker: Up & Running, 3rd Edition, April 2023

1.3 Mathematik 1

Name / engl.

Mathematik 1 / Mathematics 1

Kürzel

1.3

Verantwortlicher

Prof. Dr. Stefan Glasauer

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Wintersemester

Arbeitsaufwand / Zusammensetzung

SWS: 5, CPs: 5,

Präsenzzeit: 75 h, Selbststudium: 75 h, Gesamtaufwand: 150 h

Lehrveranstaltungen

Mathematik 1 (4 SWS)

Übungen zu Mathematik 1 (1 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Übungen

Prüfung

Prüfungsnummer

-

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Schriftliche Prüfung, 60 Minuten, keine Hilfsmittel

Inhalte des Moduls

- Logik und Mengenlehre
- Induktion und Rekursion
- Grundlagen der Analysis
- Funktionen einer Variablen
- Differenzialrechnung für Funktionen einer Variablen

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Nach Abschluss des Moduls sind Studierende in der Lage

- grundlegende mathematische Methoden und Zusammenhänge zu beschreiben.
- Probleme aus dem Bereich der Informatik mit Hilfe von mathematischen Methoden zu lösen.

Literaturliste

Teschl/Teschl: Mathematik für Informatiker 1 und 2, Springer.

Hartmann: Mathematik für Informatiker, Vieweg + Teubner.

1.4 Einführung in die Theoretische Informatik

Name / engl.

Einführung in die Theoretische Informatik / Introduction to Theoretical Computer Science

Kürzel

1.4

Verantwortlicher

Prof. Dr. Markus Degen

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Wintersemester

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 5,

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

Lehrveranstaltungen

Einführung in die Theoretische Informatik (3 SWS)

Praktikum Einführung in die Theoretische Informatik (1 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Praktikum

Prüfung

Prüfungsnummer

-

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Schriftliche Prüfung, 60 Minuten, Hilfsmittel: nicht-prog. Taschenrechner (kein Smartphone)

(10% Bonuspunkte durch Studienarbeit in Praktikum möglich)

Inhalte des Moduls

Das Modul führt in die Grundlagen der Theoretischen Informatik ein. Nach einer allgemeinen Einführung in die Teilgebiete der Informatik und der Klärung des Algorithmus-Begriffs widmet sich das Modul den Schwerpunkten

- Automatentheorie
- formale Sprachen
- Grammatiken und
- Grundlagen der Berechenbarkeit.

Das Modul legt Wert darauf zu zeigen, wie Methoden der theoretischen Informatik für Aufgabenstellungen der praktischen und technischen Informatik, wie z.B. der Entwicklung von Scannern und Parsern für formale Sprachen und Steuerungsfunktionalität für technische Systeme eingesetzt werden können.

Das Modul ist in einen Vorlesungs- und einen praktischen Teil untergliedert, in dem die Studierenden Übungsaufgaben zu den vorgestellten Inhalten lösen und Modelle und Algorithmen zu Konzepten der theoretischen Informatik entwickeln und präsentieren.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Nach Abschluss des Moduls sind Studierende in der Lage

- Algorithmen zu spezifizieren.
- die verschiedenen Automatentypen zu unterscheiden.
- deren Mächtigkeit zu beurteilen.
- Automaten anzuwenden, um ausgewählte Fragestellungen der lexikalischen Analyse und der Syntaxanalyse von Programmen oder Datenbeschreibungen zu lösen.

Sie können formale Automaten anwenden, um das Verhalten technischer Systeme zu modellieren und zu analysieren.

Durch die Beschäftigung mit der Theorie der Berechenbarkeit begegnen die Studierenden erstmals auch den Grenzen dessen, was Computer oder technische Systeme leisten können.

Durch die Beschäftigung mit Turing-Maschinen und äquivalenten Programmiermodellen, erlernen die Studierenden, welche Basiszutaten erforderlich sind, um alle algorithmisch beschreibbaren Probleme zu lösen.

Literaturliste

Socher, R.: Theoretische Grundlagen der Informatik, 3. Auflage, Hanser, 2008

Herold, H., Lurz, B.: Grundlagen der Informatik, 3. Auflage, Pearson, 2017

Schöning, U.: Theoretische Informatik - kurz gefasst, Springer, 2008

Hofmann, D. W.: Theoretische Informatik, Hanser, 2015

Karstens, U., Kleine Büning, H.: Modellierung: Grundlagen und Formale Methoden, Hanser, 2008

1.5 Rechnerstrukturen

Name / engl.

Rechnerstrukturen / Computer Structures

Kürzel

1.5

Verantwortlicher

Dr. Volodymyr Brovko

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Wintersemester

Arbeitsaufwand / Zusammensetzung

SWS: 5, CPs: 5,

Präsenzzeit: 75 h, Selbststudium: 75 h, Gesamtaufwand: 150 h

Lehrveranstaltungen

Rechnerstrukturen (4 SWS),

Praktikum Rechnerstrukturen (1 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht

Prüfung

Prüfungsnummer

-

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Schriftliche Prüfung, 60 Minuten, Hilfsmittel: nicht-prog. Taschenrechner, 2 DIN-A4-Seiten handgeschriebene Formelsammlung

Inhalte des Moduls

Grundlagen:

- Informationsdarstellung
- Binärarithmetik
- Schaltnetze
- Schaltwerke

Rechnerkomponenten:

- Maschinenbefehle
- ALU
- Hauptspeicher
- Systembus
- Prozessor
- Typische Datenwege

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

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

- Darstellung von unterschiedlichen Datentypen im Rechnerspeicher zu beschreiben.
- Grundoperationen der Binärarithmetik für Ganzzahlen, Fest- und Gleitkommazahlen zu erklären und einfache numerische Beispiele zu evaluieren.
- typische Bestandteile eines Rechners zu kennen und dessen Zusammenspiel zu erklären.
- technische Implementierung von wesentlichen Baugruppen eines Prozessors zu kennen.
- Datenpfade bei Ausführung von typischen Maschinenbefehlen zu beschreiben und Zusammenspiel von wesentlichen Prozessorbaugruppen zu analysieren.

Literaturliste

Hoffmann, D.: Grundlagen der technischen Informatik. Carl Hanser Verlag München, 2023.

Patterson, D., Hennessy, J.: Rechnerorganisation und Rechnerentwurf. De Gruyter Studium Oldenbourg, 2023.

Fertig, A.: Rechnerarchitektur Grundlagen. BoD, Norderstedt, 2018.

Hellmann, R.: Rechnerarchitektur: Einführung in den Aufbau moderner Computer. Oldenbourg, 2022.

1.6 Objektorientierte Programmierung

Name / engl.

Objektorientierte Programmierung / Object Oriented Programming

Kürzel

2.1

Verantwortlicher

Prof. Dr. Lothar Braun

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Sommersemester

Arbeitsaufwand / Zusammensetzung

SWS: 6, CPs: 10,

Präsenzzeit: 90 h, Selbststudium: 210 h, Gesamtaufwand: 300 h

Lehrveranstaltungen

Objektorientierte Programmierung (4 SWS)

Praktikum Objektorientierte Programmierung (2 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht

Prüfung

Prüfungsnummer

-

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Portfolioprüfung:

1. Elektronische Prüfung, 60 Minuten, keine Hilfsmittel
 2. Praktische Prüfung, Arbeitsumfang: 22 h
-

Ergänzende Hinweise zur Prüfungsform

Die Note für das Modul wird wie folgt aus den beiden Teilleistungen gebildet:

- Wird der zweite Teil nicht bestanden, gilt das Modul als nicht bestanden.
 - Wird der zweite Teil bestanden, wird als Modulnote die Note des ersten Teils vergeben.
-

Inhalte des Moduls

Vertiefung der Konzepte und Prinzipien einer modernen objektorientierten Programmiersprache (Java), mit Fokus auf der Entwicklung von Softwarelösungen, die für die Gestaltung und Optimierung von Anwendungen relevant sind.

Vertiefung der objektorientierten Programmierung:

- Vererbung und Polymorphie
- Interfaces und deren Anwendung
- Geschachtelte Klassen
- Meta-Programmierung und Generics
- Datenstrukturen und Container (inkl. Collections)

Weitere Programmiertechniken:

- Fehlerbehandlung und Exceptions
- Unit Tests
- Ein-/Ausgabe und Streams
- Nebenläufigkeit
- Grafische Oberflächen

Ergänzend wird ein Praktikum zur Anwendung und Vertiefung der erworbenen Kenntnisse durchgeführt. Zusätzlich unterstützt und fördert das Praktikum die Gruppenarbeit und das Selbststudium.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Nach Abschluss des Moduls sind die Studierenden in der Lage

- objektorientierte Programmierparadigmen zu beschreiben und zu erklären.
- zentrale Konzepte der objektorientierten Programmierung wie Vererbung, Polymorphie und Generics zur Lösung von Problemstellungen anzuwenden.
- die Programmierparadigmen im Rahmen der Entwicklung und von produktiver Anwendungssoftware anzuwenden.
- den Aufbau und die Verwendung von umfangreichen APIs für verschiedene Anwendungszwecke aufzuzählen und zur Lösung eigener Probleme einzusetzen.
- einfache Programme mit Fehlerbehandlung und Tests zuverlässig und strukturiert zu entwickeln.

Literaturliste

Goll, J. et al.: *Java als erste Programmiersprache*. Teubner Verlag, Stuttgart, 2016. DOI:
<https://doi.org/10.1007/978-3-658-12118-1>

Ullenboom, Christian: *Java ist auch eine Insel: Das umfassende Handbuch*. Galileo Computing, Bonn, 2019. Online-Ausgabe verfügbar unter: <http://www.tutego.com/javabuch/online.htm>.

1.7 Software Engineering

Name / engl.

Software Engineering / Software Engineering

Kürzel

2.2

Verantwortlicher

Prof. Dr. Anja Metzner

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Sommersemester

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 5,

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

Lehrveranstaltungen

Software Engineering (3 SWS)

Praktikum Software Engineering (1 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Übungen, Praktikum

Prüfung

Prüfungsnummer

-

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Schriftliche Prüfung, 60 Minuten, keine Hilfsmittel

Zusätzliche Informationen

hilfreiche Voraussetzungen

Einführung in die Programmierung

nützlich für

Software Modellierung, Angewandtes Software Engineering

Inhalte des Moduls

Die Vorlesung vermittelt grundlegende Konzepte und Methoden des Software Engineering zur systematischen Entwicklung von Softwaresystemen. Insbesondere werden Vorgehensweisen zur Bewältigung der Komplexität über den gesamten Erstellungsprozess und Lebenszyklus einer Software betrachtet.

Im Fokus der Vorlesung stehen folgende Themen:

- Einführung in Software Engineering
- Der Lebenszyklus von Software Systemen
- Vorgehensmodelle
- Planungsphase
- Definitionsphase und Requirements Engineering
- Software Designphase
- Implementierungsphase
- Verifikation und Validation: Testen von Software
- Software Wartung

Durch Übungen werden die erworbenen Kenntnisse angewandt und vertieft. Zusätzlich unterstützen und fördern die Übungen das Selbststudium.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

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

- wichtige Fachbegriffe des Software Engineering zu beschreiben.
- die Bedeutung und die Notwendigkeit von Software Engineering zu verstehen.
- Ablauf und Aufgaben des Software Engineerings zu verstehen.
- erste, ausgewählte Software Engineering Methoden anzuwenden.
- wesentliche UML-Diagramme zu verstehen und selbst zu modellieren.
- Software Architekturen zu verstehen.

Literaturliste

Begleitend zur Vorlesung:

Anja Metzner: Software Engineering - kompakt, 1. Auflage, Carl Hanser Verlag, 2020

Bernd Österreich, Axel Scheithauer: Die UML-Kurzreferenz 2.5 für die Praxis: kurz, bündig, ballastfrei, 6. Auflage, Oldenbourg Wissenschaftsverlag, 2014

Christine Rupp und SOPHISTen: Requirements-Engineering und -Management: Das Handbuch für Anforderungen in jeder Situation, 7. Auflage, Carl Hanser Verlag, 2020

Ian Sommerville: Software Engineering, 10. Auflage, Pearson Studium, Addison-Wesley, 2018

1.8 Mathematik 2

Name / engl.

Mathematik 2 / Mathematics 2

Kürzel

2.3

Verantwortlicher

Prof. Dr. Stefan Glasauer

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Sommersemester

Arbeitsaufwand / Zusammensetzung

SWS: 5, CPs: 5,

Präsenzzeit: 75 h, Selbststudium: 75 h, Gesamtaufwand: 150 h

Lehrveranstaltungen

Mathematik 2 (4 SWS)

Übungen zu Mathematik 2 (1 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Übungen

Prüfung

Prüfungsnummer

-

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Schriftliche Prüfung, 60 Minuten, Hilfsmittel: Taschenrechner

Inhalte des Moduls

- Integralrechnung
- Zahlentheorie
- Lineare Gleichungssysteme
- Matrizen und Vektoren
- Komplexe Zahlen
- Differenzialrechnung in mehreren Variablen

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Nach Abschluss des Moduls sind die Studierenden in der Lage

- mathematische Methoden in der Informatik und ihren Anwendungsgebieten zur Problemlösung anzuwenden.
- sich später benötigte mathematische Hilfsmittel auf Grundlage der Module Mathematik 1 und 2 nach Bedarf selbst anzueignen.

Literaturliste

Teschl/Teschl: Mathematik für Informatiker 1 und 2, Springer.

Hartmann: Mathematik für Informatiker, Vieweg + Teubner.

1.9 Algorithmen und Datenstrukturen

Name / engl.

Algorithmen und Datenstrukturen / Algorithms and data structures

Kürzel 2.4	Verantwortlicher Prof. Dr.-Ing. Honorary Doctor of ONPU Thorsten Schöler
----------------------	--

Lehrsprache Deutsch	Fakultät Fakultät für Informatik
-------------------------------	--

Verwendbarkeit Informatik Bachelor	Dauer / Angebot ein Semester, jeweils im Sommersemester
--	---

Arbeitsaufwand / Zusammensetzung
SWS: 5, CPs: 5,
Präsenzzeit: 75 h, Selbststudium: 75 h, Gesamtaufwand: 150 h

Lehrveranstaltungen
Algorithmen und Datenstrukturen (4 SWS),
Praktikum Algorithmen und Datenstrukturen (1 SWS)

Lehr-/Lernmethoden
Seminaristischer Unterricht, Praktikum

Prüfung

Prüfungsnummer -	Benotung Gemäß § 20 der APO in der jeweils gültigen Fassung.
----------------------------	--

Prüfungsform
Schriftliche Prüfung, 60 Minuten, Hilfsmittel: nicht-prog. Taschenrechner(kein Smartphone)
(10% Bonuspunkte durch Studienarbeit in Praktikum möglich)

Zusätzliche Informationen

hilfreiche Voraussetzungen

- Einführung in die Theoretische Informatik
 - Einführung in die Programmierung
 - Grundlegende Mathematikkenntnisse (Schulmathematik)
-

nützlich für

Alle Studiengänge der Fakultät Informatik

Inhalte des Moduls

- Einleitung
 - Informatik
 - Algorithmen und Datenstrukturen
- Kurze Einführung in Python
- Algorithmen
 - Eigenschaften
 - Darstellungsformen
 - Laufzeitanalyse
 - Implementierung
 - Datenrepräsentation
 - Grundlegende Datenstrukturen
- Sortieralgorithmen
 - Naive Sortieralgorithmen
 - Performantere Sortieralgorithmen
- Suchalgorithmen
 - Einleitung
 - Bäume, z. B.
 - * Binärbäume
 - * B-Bäume
 - * Hashing
 - * Tries
- Graphalgorithmen
 - Grundlagen/Graphen
 - Ausgewählte Graphalgorithmen, z. B.
 - * Breiten- und Tiefensuche
 - * Backtracking
 - * (Topologisches Sortieren)
 - * Kürzeste Wege
 - * Spannbäume
 - * Maximaler Fluss
- Stringmatching

- Grundlagen
- Ausgewählte Verfahren, z. B.
 - * Primitiver Algorithmus
 - * Endliche Automaten
 - * Knuth-Morris-Pratt
 - * Boyer-Moore
 - * (Rabin-Karp)
 - * (Shift-Or)
 - * (Benchmarks)
- Schwere Probleme, Heuristiken und statistische Algorithmen
 - Grundlagen
 - Ausgewählte Verfahren, z. B.
 - * Traveling Salesman Problem
 - * Rucksackproblem
 - * Ähnlichkeitsanalyse
 - * Klassifizierung

Im Praktikumsteil werden Übungsaufgaben zu den wesentlichen in der Vorlesung systematisch vorgestellten Algorithmen und Datenstrukturen gemeinsam erarbeitet.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Nach Abschluss des Moduls sind Studierende in der Lage, grundlegende Algorithmen und Datenstrukturen der Informatik zu verstehen, einzuordnen und zu verwenden. Die Studierenden beherrschen die theoretischen Grundlagen von Algorithmen und Datenstrukturen.

Die Studierenden erlangen Kenntnisse grundlegender Datenstrukturen und Verarbeitungstechniken/ Algorithmen und die Fähigkeit, sie anzuwenden. Dazu zählen Komplexitätsanalyse, Laufzeitanalyse, Sortierverfahren, Suchverfahren, graphenbasierte Verfahren, Stringmatching sowie Verfahren zur Lösung schwerer Probleme wie z. B. Heuristiken und statistische Algorithmen.

Literaturliste

- Medientechnisches Wissen: Band 2 Informatik, Programmieren, Kybernetik. De Gruyter Oldenbourg, 2018. doi: 10.1515/9783110496253.
Verfügbar unter: <http://www.degruyter.com/document/doi/10.1515/9783110496253/html>.
(Zugegriffen: 6. April 2022)
- T. Häberlein:** Praktische Algorithmik mit Python. München: Oldenbourg, 2012.
- J. Guttag:** Introduction to Computation and Programming Using Python, third edition: With Application to Computational Modeling and Understanding Data. The MIT Press, 2021.
- I. Ahmad und S. Nikpoor:** 50 Algorithms Every Programmer Should Know: Tackle computer science challenges with classic to modern algorithms in machine learning, software design, ... algorithmic solutions for real-world problems, 2. Aufl. Packt Publishing, 2023.
- H. Herold, B. Lurz, und J. Wohlrab:** Grundlagen der Informatik, Auflage: 2., Aktualisierte Auflage. München: Pearson Studium, 2012.
- G. Saake und K.-U. Sattler:** Algorithmen und Datenstrukturen: eine Einführung mit Java. Heidelberg: dpunkt, 2014.
- U. Schöning:** Ideen der Informatik: Grundlegende Modelle und Konzepte der Theoretischen Informatik, Korrigierte Auflage. München: Oldenbourg Wissenschaftsverlag, 2008.

1.10 Rechnerarchitektur

Name / engl.

Rechnerarchitektur / Computer Architecture

Kürzel

2.5

Verantwortlicher

Prof. Dr. Michael Strohmeier

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Sommersemester

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 5,

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

Lehrveranstaltungen

Rechnerarchitektur (3 SWS)

Übungen zu Rechnerarchitektur (1 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht und begleitende Praktika zur Anwendung und Vertiefung der erworbenen Kenntnisse.

Prüfung

Prüfungsnummer

-

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Klausur, 60 Minuten, Hilfsmittel: Taschenrechner, 2 DIN-A4-Seiten handgeschriebene Formelsammlung

Zusätzliche Informationen

hilfreiche Voraussetzungen

Rechnerstrukturen

Inhalte des Moduls

Die Veranstaltung vermittelt grundlegende Kenntnisse zu modernen Rechnerarchitekturen und deren Organisation. Im Einzelnen werden folgende Themen behandelt:

- Leistungsbewertung von Rechnersystemen
- Universalrechner und grundlegende Architekturen
- Leit- und Rechenwerke
- Befehlssatzarchitekturen (ISA)
- Pipelines und Parallelität auf Instruktions-, Daten- und Threadebene
- Speicherhierarchien und virtueller Speicher
- Trends und Entwicklungen in der Rechnerarchitektur

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Nach Abschluss des Moduls sind Studierende in der Lage:

- grundlegende Leistungsbewertungen von Rechnersystemen durchzuführen.
- die grundlegenden Prinzipien moderner Rechnerarchitekturen zu beschreiben und zu vergleichen.
- unterschiedliche Formen der Parallelisierung zu bewerten und geeignete Ansätze für konkrete Probleme auszuwählen.
- aktuelle Trends und Technologien zu beschreiben.

Literaturliste

Patterson D.A., Hennessy J.L. Computer Organization and Design RISC-V Edition: The Hardware/Software Interface, Revised 2nd Edition, Morgan Kaufmann, 2021

Hennessy J.L., Patterson D.A. Computer Architecture: A Quantitative Approach, 6th Edition, Morgan Kaufmann, 2017

Hennessy J.L., Patterson D.A. A New Golden Age for Computer Architecture. In: Communications of the ACM 62, 2 Jg. (2019), S. 48-60

2 Vertiefungsphase

2.1 Fortgeschrittene Programmiertechniken

Name / engl.

Fortgeschrittene Programmiertechniken / *Advanced Programming Techniques*

Kürzel

3.1

Verantwortlicher

Prof. Dr. Peter Rösch

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Wintersemester

Arbeitsaufwand / Zusammensetzung

SWS: 6, CPs: 10,

Präsenzzeit: 90 h, Selbststudium: 210 h, Gesamtaufwand: 300 h

Lehrveranstaltungen

Fortgeschrittene Programmiertechniken (4 SWS),

Praktikum Fortgeschrittene Programmiertechniken (2 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Praktikum

Prüfung

Prüfungsnummer

-

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Portfolioprüfung:

1. Schriftliche Prüfung, 60 Minuten, keine Hilfsmittel
 2. Praktische Prüfung, Arbeitsumfang: 22 h
-

Ergänzende Hinweise zur Prüfungsform

Die Note für das Modul wird wie folgt aus den beiden Teilleistungen gebildet:

- Wird der zweite Teil nicht bestanden, gilt das Modul als nicht bestanden.
 - Wird der zweite Teil bestanden, wird als Modulnote die Note des ersten Teils vergeben.
-

Zusätzliche Informationen

hilfreiche Voraussetzungen

- Objektorientierte Softwareentwicklung
 - Linux (Grundlagen)
-

Inhalte des Moduls

Systematisch vom Problem zur Software

- Einführung in Python
- Entwicklung von Prototypen in Notebooks
- Implementierung automatisch ausführbarer Tests
- Automatisierte Erzeugung von Dokumentation
- Eigene Module und Pakete
- Grafische Benutzerschnittstellen

Fortgeschrittene Programmier Techniken mit Python:

- Nebenläufigkeit
- Verteilte Anwendungen
- Systematische Optimierung
- Wissenschaftliche Anwendungen

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

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

- die syntaktischen Konstrukte der Programmiersprache Python zu beschreiben.
- automatisch erzeugten Quellcode bezüglich Effizienz und Qualität zu klassifizieren.
- den Ressourcenverbrauch von Anwendungen zu analysieren und systematisch zu reduzieren.
- Bibliotheken, die für die Lösung von Teilproblemen geeignet sind, zu identifizieren.
- Probleme mittlerer Komplexität in Teilaufgaben zu zerlegen und effizient im Team zu lösen.
- Software-Komponenten zu entwickeln, zu testen, zu dokumentieren und als installierbare Pakete zur Verfügung zu stellen.

Literaturliste

Tobias Häberlein: Programmieren mit Python. Springer (2024)

Paul Barry: Python von Kopf bis Fuß, 3. Auflage. dpunkt (2024)

Michal Jaworski, Tarek Ziadé: Expert Python Programming, 4th Ed., Packt (2021)

Python-Homepage: <https://www.python.org/>

2.2 Softwaremodellierung

Name / engl.

Softwaremodellierung / Software Modeling

Kürzel

3.2

Verantwortlicher

Prof. Dr. Phillip Heidegger

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Wintersemester

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 5,

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

Lehrveranstaltungen

Softwaremodellierung (3 SWS)

Praktikum Softwaremodellierung (1 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Praktikum

Prüfung

Prüfungsnummer

-

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Schriftliche Prüfung, 60 Minuten, keine Hilfsmittel

Zusätzliche Informationen

hilfreiche Voraussetzungen

Modul Software Engineering, Modul Einführung in die Programmierung

Inhalte des Moduls

Die Vorlesung Software Modellierung konzentriert sich auf die Vermittlung der in der Entwurfsphase der Softwareentwicklung notwendigen Techniken und Methoden. Hierzu werden zuerst Entwurfsprinzipien von Software vorgestellt.

Ein weiterer Aspekt der Veranstaltung ist die Modellierung von Software mithilfe von UML. Es werden die wichtigsten UML-Diagramme besprochen und deren Einsatz in der Softwareentwicklung erläutert. Anschließend werden dann unter Zuhilfenahme der UML häufig eingesetzte Entwurfsmuster vorgestellt und mit den zu Beginn der Veranstaltung besprochenen Entwurfsprinzipien in Verbindung gebracht.

Anhand eines Miniprojektes werden diese Techniken angewendet. Dabei werden auch Themen zur agilen Arbeitsorganisation und -planung eingeführt sowie Aspekte der Teamarbeit erläutert.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Die Studierenden

- verstehen die grundlegenden Softwareentwurfsprinzipien, können diese erkennen und anwenden.
- können Sachverhalte mit UML-Diagrammen beschreiben.
- kennen wichtige Entwurfsmuster und können diese anwenden.
- können Arbeitsschritte aufteilen und im Team entwickeln.
- können Softwareentwicklungstechniken in realistischen Szenarien einsetzen.

Literaturliste

- Chris Rupp und Stefan Queins, UML 2 glasklar: Praxiswissen für die UML-Modellierung, 2012, ISBN-13: 978-3446430570
- Florian Siebler , Design Patterns mit Java, Eine EINFÜHRUNG in ENTWURFSMUSTER, 2014, Print-ISBN: 978-3-446-43616-9, E-Book-ISBN: 978-3-446-44111-8
- Erich Gamma, Richard Helm, Ralph Johnson und John Vlissides, Design Patterns. Elements of Reusable Object-Oriented Software, 2015, ISBN: 0-201-63361-2
- Ludewig und Lichter, Software Engineering: Grundlagen, Menschen, Prozesse, Techniken; dpunkt.verlag 3. korrigierte Auflage 2013, ISBN-13: 978-3864900921
- Boris Gloger, Scrum: Produkte zuverlässig und schnell entwickeln, Carl Hanser Verlag GmbH, 2008, ISBN-13: 978-3446414952
- Ester Derby, Diana Larson, Agile Retrospectives: Making Good Teams Great, O'Reilly 2004, ISBN-13: 978-0977616640

2.3 Statistik

Name / engl.

Statistik / Statistics

Kürzel

3.3

Verantwortlicher

Dr. Torsten Straßer

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Wintersemester

Arbeitsaufwand / Zusammensetzung

SWS: 5, CPs: 5,

Präsenzzeit: 75 h, Selbststudium: 75 h, Gesamtaufwand: 150 h

Lehrveranstaltungen

Statistik (5 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Übungen

Prüfung

Prüfungsnummer

-

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Schriftliche Prüfung, 90 Minuten, Hilfsmittel: Skript, eigene Notizen, Statistikbuch, Notebook, Tablet, Statistiksoftware (z. B. SAS JMP, R, o.ä.), Internetzugang

Zusätzliche Informationen

hilfreiche Voraussetzungen

- Lineare Algebra und Finanzmathematik
- Analysis
- Einführung in die Programmierung
- Programmierung in der Wirtschaftsinformatik

Grundkenntnisse in Mathematik (aus 1. und 2. Semester)

Grundlegende Programmierkenntnisse (aus 1. und 2. Semester)

Inhalte des Moduls

- Einführung in die Statistik:
 - Rolle der Statistik in Wissenschaft und Forschung, insbesondere in Informatik und Data Science
 - Datentypen und Messskalen
 - Grundbegriffe: Grundgesamtheit, Stichprobe, Zufallsexperiment
- Deskriptive Statistik:
 - Lage- und Streuungsmaße (Mittelwert, Median, Varianz etc.)
 - Datenvisualisierung (Histogramme, Boxplots, Scatterplots)
- Wahrscheinlichkeitstheorie:
 - Grundbegriffe der Wahrscheinlichkeitsrechnung
 - Kombinatorik
 - Diskrete und stetige Verteilungen (z. B. Binomial-, Normalverteilung)
- Schließende Statistik:
 - Stichproben und Stichprobenverteilungen
 - Konfidenzintervalle
 - Hypothesentests (u.a. z-Test, t-Test, ANOVA)
- Korrelation und Regression:
 - Zusammenhangsmaße (z. B. Pearson-Korrelation)
 - Lineare Regression
 - Einführung in multiple Regression
- Statistische Software:
 - Einführung in Softwarepakete zur Datenanalyse (SAS JMP)

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Die Studierenden:

- verstehen grundlegende statistische Konzepte und Methoden.
- sind in der Lage, Daten zu beschreiben, zu visualisieren und zu analysieren.
- können Wahrscheinlichkeiten und statistische Zusammenhänge berechnen.
- beherrschen statistische Verfahren wie Hypothesentests und Regressionsanalysen.
- sind in der Lage, quantitative Anwendungsprobleme in der Informatik vor dem Hintergrund der in der Vorlesung erlernten statistischen Methoden zu bewerten, zu modellieren, Lösungsansätze zu entwickeln und die Ergebnisse korrekt und kritisch zu interpretieren.
- verfügen über hinreichende Kenntnisse in einem statistischen Softwarepaket, um die erlernten Methoden in der Praxis anwenden zu können.
- können statistische Ergebnisse interpretieren und kritisch bewerten.

Literaturliste

Meintrup, David: Angewandte Statistik: Eine Einführung mit JMP, 1. Auflage: CreateSpace Open Publishing Platform. ISBN 978-1-9816-6989-9

2.4 Daten- & Informationsanalyse

Name / engl.

Daten- & Informationsanalyse / *Data and Information Analysis*

Kürzel

3.4

Verantwortlicher

Prof. Matthias Kolonko, Ph.D. (ONPU)

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Wintersemester

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 5,

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

Lehrveranstaltungen

Daten- & Informationsanalyse (3 SWS)

Praktikum Daten- & Informationsanalyse (1 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Übungen, Praktikum

Prüfung

Prüfungsnummer

-

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Portfolioprüfung:

1. Schriftliche Prüfung, 60 Minuten, keine Hilfsmittel
 2. Praktische Prüfung, Arbeitsumfang: max. 11h
-

Ergänzende Hinweise zur Prüfungsform

Die Note für das Modul wird wie folgt aus den beiden Teilleistungen gebildet:

- Wird der zweite Teil nicht bestanden, gilt das Modul als nicht bestanden.
 - Wird der zweite Teil bestanden, wird als Modulnote die Note des ersten Teils vergeben.
-

Inhalte des Moduls

Die Vorlesung befasst sich mit der Analyse und Darstellung von Datenstrukturen anhand der vorliegenden Umstände in der realen Welt.

Kernpunkt hierbei ist die Entwicklung anhand des 3-stufigen Entwicklungsprozesses für Datenbanken mit der Unterscheidung in konzeptuelles, logisches und physisches Datenmodell.

Die Vorlesung legt dabei den Schwerpunkt auf die konzeptuelle Modellierung und befasst sich hierbei insbesondere mit der Informationsanalyse aus Gegebenheiten der realen Welt und der daraus resultierenden semantischen Datenmodellierung.

In dem Zuge wird auch der Begriff der Kommunikation kurz beleuchtet, der bei der gemeinsamen Entwicklung zwischen Fach- und Technikexperten erforderlich ist. Es werden insbesondere Merkmale natürlicher Sprache aufgezeigt, welche in der Kommunikation zu Fehlern führen können.

Der so erstellte systemunabhängige und anomaliefreie Entwurf wird anschließend in einen relationalen Entwurf als Beispiel eines logischen Datenmodells überführt.

Zu diesem Zweck werden Regeln und Techniken aufgezeigt, anhand derer eine Ableitung durch einen fest vorgegebenen und einfachen Prozess vom konzeptuellen Modell vorgenommen werden kann.

Zuletzt wird vertiefend auf die Normalformtheorie eingegangen. Hierbei werden sowohl praktische als auch theoretische Aspekte beleuchtet und das Vorgehen einer Normalformenanalyse vorgestellt.

Die Inhalte werden anhand eines Praktikums in Kleingruppen durch die Teilnehmenden anhand eines selbst gewählten Themas geübt und verinnerlicht.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Nach erfolgreicher Absolvierung des Moduls sind die Studierenden in der Lage

- Kommunikation als Mittel der Informationsübertragung zu verstehen.
- Probleme bei der Kommunikation zu nennen.
- Lösungsansätze für Kommunikationsprobleme aufzuzeigen.
- den Begriff der semantischen Datenmodellierung zu erklären.
- komplexe Aussagen in einfache, semantisch irreduzible Sätze zu zerlegen.
- eine Analyse der vorliegenden Informationen aus der realen Welt durchzuführen.
- ein systemunabhängiges und anomaliefreies konzeptuelles Datenmodell zu erstellen.
- ein logisches Datenmodell durch Ableitung von einem konzeptuellen Modell herzuleiten.
- die verschiedenen Normalformen nennen und deren Voraussetzungen beschreiben.
- die Analyse und Normalisierung eines logischen Datenmodells anhand der Normalformtheorie anwenden.

Literaturliste

- R. Elmasri, S. B. Navathe: *Fundamentals of Database Systems* (Pearson 2020, ISBN: 1-292-09761-2)
- S. Müllenbach, L. Kern-Bausch, M. Kolonko: Conceptual Modeling Language AGILA MOD
in Herald of Advanced Information Technology, vol. 2, no. 4, pp. 246-258, Dez. 2019
(ISSN: 2663-0176 – DOI: 10.15276/hait.04.2019.1)
- M. Kolonko, S. Müllenbach, E. Arsirii, B. Trofymov: *Extensions to the Conceptual Modeling Language AGILA MOD*
in Proceedings of the VI. Ukrainian-German conference „Informatics. Culture. Technology“, Odessa, Sept. 2018, pp. 38-39
- L. Kern-Bausch, M. Jeckle: Informationsmodellierung und logischer Datenbankentwurf, Kapitel 14.2
in Taschenbuch der Informatik (U. Schneider und D. Werner), 4. Auflage, Fachbuchverlag Leipzig im Carl Hanser Verlag, 2001,
ISBN: 3-446-21753-3
- P. Sauer: *Informationsmodellierung, Kapitel 2*
in Taschenbuch Datenbanken (T. Kudraß), 2. Auflage, Fachbuchverlag Leipzig im Carl Hanser Verlag, 2015,
ISBN: 978-3-446-43508-7
- Vorlesungsunterlagen von Prof. Dr. Sabine Müllenbach unter <https://ohs.informatik.hs-augsburg.de:4443/web/bine>
(Anmeldung mit RZ-Login)

2.5 Computernetzwerke

Name / engl.

Computernetzwerke / Computer Networks

Kürzel

3.5

Verantwortlicher

Prof. Dr. Rolf Winter

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Wintersemester

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 5,

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

Lehrveranstaltungen

Computernetzwerke (3 SWS)

Praktikum Computernetzwerke (1 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Praktikum

Prüfung

Prüfungsnummer

-

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Schriftliche Prüfung, 60 Minuten, Hilfsmittel: Taschenrechner

Inhalte des Moduls

Funktionsweise und Aufbau der Internet-Architektur und seiner Prinzipien und Protokolle insbesondere:

- Protokolle der Anwendungsschicht (wie HTTP und DNS)
- Transport-Protokolle (wie TCP und UDP)
- Routing-Protokolle (link state und distance vector)
- Protokolle der Sicherungsschicht (z.B. Ethernet)
- Arbeitsweise von Kernkomponenten des Internets (Switches, CDNs, NAT, uvm.)
- Schlüsselprinzipien des Internets (Zuverlässige Datenübertragung, Staukontrolle etc.)
- Umgang mit Standardwerkzeugen (Software) im Bereich Netzwerke

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Die Studierenden sind in der Lage

- die Schlüsselprotokolle des Internets und deren Aufgaben und Funktionsweise im Detail zu erklären.
- die Funktionen der Internet-Architektur und wie und wo diese im Netz implementiert sind, zu verstehen.
- die komplexen Zusammenhänge zwischen Protokollen und Mechanismen im Internet zu beschreiben.
- ihr erlerntes Wissen auch praktisch bei der Entwicklung von vernetzten Anwendungen oder der Einrichtung und Wartung von Netzen einsetzen.
- durch das Praktikum befähigt, mit Standardwerkzeugen Anwendungen und Netze zu analysieren.

Literaturliste

Es gibt eine Vielzahl an Quellen in Print als auch online, die grundsätzlich geeignet sind. Die Vorlesung hält sich aber in Inhalt und Struktur an folgendes Buch:

Kurose, J.F./ Ross, K.W.: Computernetzwerke, 6. Auflage, Pearson Studium, 3/2014, ca. 900 Seiten, ISBN 978-3-8689-4237-8

2.6 Datenbanksysteme

Name / engl.

Datenbanksysteme / Database Systems

Kürzel

4.1

Verantwortlicher

Prof. Matthias Kolonko, Ph.D. (ONPU)

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Sommersemester

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 5,

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

Lehrveranstaltungen

Datenbanksysteme (3 SWS),

Praktikum Datenbanksysteme (1 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Übungen, Praktikum

Prüfung

Prüfungsnummer

-

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Portfolioprüfung:

1. Schriftliche Prüfung, 60 Minuten, keine Hilfsmittel
 2. Praktische Prüfung, Arbeitsumfang: max. 11h
-

Ergänzende Hinweise zur Prüfungsform

Die Note für das Modul wird wie folgt aus den beiden Teilleistungen gebildet:

- Wird der zweite Teil nicht bestanden, gilt das Modul als nicht bestanden.
 - Wird der zweite Teil bestanden, wird als Modulnote die Note des ersten Teils vergeben.
-

Zusätzliche Informationen

hilfreiche Voraussetzungen

- Daten- und Informationsanalyse
-

Inhalte des Moduls

Die Vorlesung befasst sich mit Datenbanksystemen im Allgemeinen und relationalen Datenbanksystemen im Speziellen.

Dabei werden zunächst grundlegende Überlegungen für den Einsatz von Datenbanksystemen besprochen und anschließend allgemeine Architekturprinzipien eines Datenbanksystems beleuchtet.

Anschließend richtet sich der Fokus auf relationale Datenbanksysteme. Hierbei werden die grundlegenden strukturellen Aspekte und Operationen vorgestellt. Ebenso werden physische Aspekte der Datenspeicherung betrachtet.

Anschließend wird deren Umsetzung anhand der Structured Query Language (SQL) aufgezeigt.

Desweiteren befasst sich die Vorlesung mit verarbeitungstechnischen Aspekten wie Transaktionen und Optimierungen. Hierbei wird explizit auf das ACID-Prinzip eingegangen und dessen Vor- und Nachteile - insbesondere mit Blick auf Sperren - beleuchtet. Daneben werden auch alternative Konzepte kurz betrachtet.

Die Architektur eines Datenbank-Management-Systems und geeignete physische Datenstrukturen werden an einem verbreiteten RDBMS aufgezeigt.

Es wird eine Instanz einer relationalen Datenbank für SQL-Übungen zur Verfügung gestellt. Im Praktikum werden die Realisierung und der Einsatz relationaler Datenbanken geübt.

Die dafür nötigen Strukturen können entweder anhand der Übungen aus dem *Praktikum Daten- und Informationsanalyse* erstellt werden, sofern dies bereits besucht wurde. Andernfalls werden passende Strukturen zur Verfügung gestellt.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Nach erfolgreicher Absolvierung des Moduls können die Studierenden

- Gründe für den Einsatz eines Datenbanksystems nennen.
- die grundlegende Architektur und Funktionsweise eines Datenbanksystems beschreiben.
- die grundlegenden relationalen Operationen erklären.
- SQL in die verschiedenen Teilsprachen eines Datenbanksystems einordnen.
- das ACID-Prinzip beschreiben.
- Datenstrukturen und Anfragen an eine relationale Datenbanke mittels SQL (DDL & DML) realisieren.

Literaturliste

- R. Elmasri, S. B. Navathe: *Fundamentals of Database Systems* (Pearson 2020, ISBN: 1-292-09761-2)
- M. Kofler: *Datenbanksysteme: Das umfassende Lehrbuch* (Rheinwerk Verlag 2022, ISBN: 9783836284226)
- W. Lemahieu, S. vanden Broucke, B. Baesens: *Principles of Database Management* (Cambridge University Press 2018, ISBN: 978-1107186125)
- Vorlesungsunterlagen von Prof. Dr. Sabine Müllenbach unter <https://ohs.informatik.hsaugsburg.de:4443/web/bine> (Anmeldung mit RZ-Login)

2.7 Angewandtes Software Engineering

Name / engl.

Angewandtes Software Engineering / Applied Software Engineering

Kürzel

4.2

Verantwortlicher

Prof. Dr. Phillip Heidegger

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Sommersemester

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 5,

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

Lehrveranstaltungen

Angewandtes Software Engineering (2 SWS),

Praktikum Angewandtes Software Engineering (2 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Praktikum

Prüfung

Prüfungsnummer

-

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Praktische Prüfung (Umfang: 22 Stunden)

Zusätzliche Informationen

hilfreiche Voraussetzungen

Modul Softwaremodellierung (empfohlen), Modul Objektorientierte Programmierung (empfohlen)

Inhalte des Moduls

Den Schwerpunkt dieses Moduls bilden verschiedene Verfahren, Qualität von Software sicherzustellen. Hierfür wird zuerst der Begriff der Qualität eingehend erklärt.

Der Zusammenhang zwischen Architektur und Qualität von Software wird besprochen und im Praktikum durch Diskussionen vertieft. Zwei unterschiedliche Sichten auf Architektur, die Macro- und die Mikroarchitektur werden vorgestellt. Um beide Sichten besser greifbar zu machen werden aus beiden Perspektiven einige Themen ausführlicher behandelt.

Bei der Makroarchitektur werden u.a. Architekturpattern behandelt. Im Bereich Mikroarchitektur wird der Unterschied zwischen synchroner und asynchroner Programmierung und die Auswirkungen auf die Softwareentwicklung besprochen.

Außerdem wird in dem Modul behandelt, wie Dokumentation von Architektur den Softwareentwicklungsprozess unterstützen kann.

Es wird dargestellt wie mithilfe von unterschiedlichen Testverfahren die Qualität von Software verbessert werden kann. Gütekriterien für Softwaretests werden behandelt.

Als letztes Thema behandelt das Modul die Frage, wie während der Entwicklung aufgestaute technische Schulden beseitigt werden können (Refactoring).

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Die Studierenden können

- die Bedeutung des Begriffs Qualität von Software erläutern.
- die Bedeutung des Begriffs Architektur von Software erläutern.
- den Zusammenhang zwischen Qualität und Architektur darlegen.
- Mikro- und Makroarchitektur unterscheiden.
- Architekturpattern in Softwareprojekten einsetzen und entscheiden, welche Pattern in welchen Szenarien Sinn machen.
- asynchron arbeitende Software entwickeln.
- Vor- und Nachteil von synchroner und asynchroner Software einschätzen.
- die Architektur von Software dokumentieren.
- mithilfe von Tests die Qualität von Software überprüfen und festgelegte Forderungen an die Qualität von Software sicherstellen.
- die Güte von Tests für Softwaresysteme einschätzen.
- Methoden anwenden, um Software strukturell zu verbessern.

Literaturliste

Fowler, 1999: Refactoring, Improving the Design of Existing Code, ISBN-13: 978-0-201-48567-7

Liggemeyer, 2009: Software-Qualität: Testen, Analysieren und Verifizieren von Software, ISBN-13: 978-3-827-42056-5

Ludewig und Lichter, 2013: Software Engineering: Grundlagen, Menschen, Prozesse, Techniken, ISBN-13: 978-3-864-90092-1

Myers, Sandler und Badgett, 2011: The Art of Software Testing, ISBN-13: 978-1-118-03196-4

Vogel u.a., 2009: Software-Architektur: Grundlagen – Konzepte – Praxis, ISBN-13: 978-3-827-41933-0

Zeller, 2009: Why Programs Fail, ISBN-13: 978-3-898-64620-8

Zörner, 2015: Software Architekturen Dokumentieren und Kommunizieren, ISBN-13: 978-3-446-44348-8

2.8 Sichere Softwareentwicklung

Name / engl.

Sichere Softwareentwicklung / Secure Software Development

Kürzel

4.3

Verantwortlicher

Prof. Dr. Lothar Braun

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Sommersemester

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 5,

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

Lehrveranstaltungen

Sichere Softwareentwicklung (2 SWS),

Praktikum Sichere Softwareentwicklung (2 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Praktikum

Prüfung

Prüfungsnummer

-

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Schriftliche Prüfung, 60 Minuten, keine Hilfsmittel

Inhalte des Moduls

Inhalt des Moduls ist die sichere Softwareentwicklung. Hierfür behandelt das Modul folgende Inhalte:

- Grundbegriffe
- Security by Design und Secure Development Lifecycle (SDL)
- Risiko- und Bedrohungsanalysen
- Schwachstellen und Bewertungen von Schwachstellen
- Vertiefung: Web-Schwachstellen
- Vulnerability Management und Third-Party-Komponenten
- Authentifizierung und Kryptographie
- Security Testing

Die behandelten Konzepte werden im Praktikum am Beispiel einer Web-Anwendung praktisch umgesetzt.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Nach Abschluss des Moduls sind Studierende in der Lage

- die relevanten Grundbegriffe der sicheren Softwareentwicklung zu beschreiben.
- die Komponenten eines SDL-Prozesses zu beschreiben und die wesentlichen Methoden anzuwenden.
- einfache Risiko- und Bedrohungsanalysen durchzuführen.
- typische Schwachstellen zu benennen und zu beschreiben und mittels gängiger Bewertungsmethoden zu klassifizieren.
- Schwachstellen im Web-Umfeld zu erkennen und in Implementierungen zu vermeiden.

Literaturliste

Aktuelle Literatur wird im Rahmen der Vorlesung bekannt gegeben.

2.9 Projektarbeit 1

Name / engl.

Projektarbeit 1 / Project work 1

Kürzel

4.4

Verantwortlicher

Studiengangsleiter

Lehrsprache

Das Modul wird in deutscher und englischer Sprache unterrichtet.

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Sommersemester

Arbeitsaufwand / Zusammensetzung

SWS: 8, CPs: 10,

Präsenzzeit: 120 h, Selbststudium: 180 h, Gesamtaufwand: 300 h

Lehrveranstaltungen

Projektarbeit 1 (8 SWS)

Lehr-/Lernmethoden

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

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

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

Prüfung

Prüfungsnummer

-

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Portfolioprüfung:

1. Studienarbeit, 10-15 Seiten, 80%
 2. Präsentation, 20-30 Minuten, 20%
-

Zusätzliche Informationen

hilfreiche Voraussetzungen

Grundlagen des Softwareengineering und Objektorientierte Programmieretechniken

Inhalte des Moduls

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

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

Zum Beginn des Moduls findet ein Seminar zur Vermittlung der Grundlagen der Teamarbeit und dem Umgang mit Teamkonflikten statt. Alle Mitglieder der Gruppen sollen zu Beginn der Arbeit ein Minimum an Kenntnissen von Teamprozessen haben, um den eigenen Lernerfolg und den Lernerfolg der Teammitglieder sicherzustellen. Aus diesem Grund ist die Teilnahme an dem Seminar verpflichtend.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

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

- Teamprozesse zu verstehen und typische Teamkonflikte zu lösen.
- Software- und Hardware-Entwicklungsprojekte im Team hinsichtlich Zeit, Aufwände und Ressourcen zu planen und durchzuführen.
- Agile oder klassische Projektmanagementmethoden praktisch anzuwenden.
- Auswahl geeigneter Methoden und selbstständiges Erlernen neuer Techniken.
- Projektergebnisse verständlich zu dokumentieren und ansprechend zu präsentieren.

Literaturliste

Projektspezifische Literatur wird vom Betreuer vor Beginn des Projektes bekanntgegeben.

2.10 Einführung in das maschinelle Lernen

Name / engl.

Einführung in das maschinelle Lernen / *Basics of machine learning*

Kürzel

5.1

Verantwortlicher

Prof. Dr. Alexandra Teynor

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Wintersemester

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 5,

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

Lehrveranstaltungen

Grundlagen des maschinellen Lernens (2 SWS),

Praktikum Grundlagen des maschinellen Lernens (2 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Praktikum

Prüfung

Prüfungsnummer

-

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Schriftliche Prüfung, 60 Minuten, Hilfsmittel: Taschenrechner

Zusätzliche Informationen

hilfreiche Voraussetzungen

Einführung in die Programmierung, Einführung in die theoretische Informatik

Inhalte des Moduls

- Arten des Maschinellen Lernens
- Entwicklungszyklus für ME-Systeme
- Datenaufbereitung (Missing Values, Imbalanced Data, Outlier)
- Mathematische Betrachtung von Ähnlichkeit
- Klassifikationsverfahren
 - Naive Bayes
 - K-nearest Neighbor
 - Lineare Regression, Lineare Klassifikation
 - SVM
 - Perzeptron, Grundlagen und Anwendung Neuronaler Netze
 - Feedforward Neural Networks (Multi-Layer Perceptron),
- Clusteringverfahren (z.B. Agglomeratives Clustering, K-Means, DBSCAN)
- Trainings- und Testmethoden
 - Hebbian Learning
 - Gradientenabstiegsverfahren und Backpropagation
 - Train-/Validation-/Test Split
 - Kreuzvalidierung
 - Bias-Varaince Trade Off
 - Overfitting
- Evaluierung
 - Evaluationsmaße (Korrelation, Recall/Precision, Sens/Spec, ROC, AUC, UAR, etc.)
 - Qualitative Evaluierung
 - Interpretable and Explainable AI
 - Ethik
- Tools und Frameworks

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Nach Abschluss des Moduls sind Studierende in der Lage, folgende Kenntnisse, Fähigkeiten und Kompetenzen anzuwenden:

- Kenntnisse:
 - Basistechniken des maschinellen Lernens
 - Klassifikatoren
 - Clustering-Algorithmen
 - Evaluationsmaße und Testing
 - typische Anwendungsfälle für den Einsatz von ML
- Fähigkeiten:
 - Daten für Mustererkennungssysteme aufzubereiten.
 - geeignete Merkmale für die Weiterverarbeitung auszuwählen, zu extrahieren und/oder zu kombinieren.
 - für gegebene Klassifikationsprobleme geeignete Klassifikatoren auszuwählen und anzuwenden.
 - die Leistungsfähigkeit von Mustererkennungssystemen auf Grund von anerkannten Gütemaßen zu vergleichen.
- Kompetenzen:
 - auf Grundlage vorgegebener Literatur Algorithmen selbständig zu erarbeiten, diese anzuwenden und deren Eigenschaften zu analysieren.
 - eigene Mustererkennungssysteme für industrielle Anwendungen mit zeigemäßen Algorithmen und Softwarepaketen zu entwickeln.

Literaturliste

Wird in der Vorlesung bekannt gegeben

2.11 Projektarbeit 2

Name / engl.

Projektarbeit 2 / Project Work 2

Kürzel

6.1

Verantwortlicher

Studiengangsleiter

Lehrsprache

Das Modul wird in deutscher und englischer Sprache unterrichtet.

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Sommersemester

Arbeitsaufwand / Zusammensetzung

SWS: 8, CPs: 10,

Präsenzzeit: 120 h, Selbststudium: 180 h, Gesamtaufwand: 300 h

Lehrveranstaltungen

Projektarbeit 2 (8 SWS)

Lehr-/Lernmethoden

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

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

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

Prüfung

Prüfungsnummer

-

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Portfolioprüfung:

1. Studienarbeit, 10-15 Seiten, 80%
 2. Präsentation, 20-30 Minuten, 20%
-

Zusätzliche Informationen

hilfreiche Voraussetzungen

Solide Kenntnisse aus den wichtigsten Themenbereichen der Informatik, wie z.B. Algorithmen und Datenstrukturen, Programmieren, Datenbanken, Datenkommunikation, Rechnerstrukturen und Betriebssysteme. Die erworbenen Kenntnisse sollten bereits in der ersten Projektarbeit praktisch angewendet worden sein.

Inhalte des Moduls

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

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

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

- Teamprozesse zu verstehen und typische Teamkonflikte zu lösen.
- Software- und Hardware-Entwicklungsprojekte im Team hinsichtlich Zeit, Aufwände und Ressourcen zu planen und durchzuführen.
- Agile oder klassische Projektmanagementmethoden praktisch anzuwenden.
- Auswahl geeigneter Methoden und selbstständiges Erlernen neuer Techniken.
- Projektergebnisse verständlich zu dokumentieren und ansprechend zu präsentieren.

Literaturliste

Projektspezifische Literatur wird vom Betreuer vor Beginn des Projektes bekanntgegeben.

2.12 Betriebssysteme

Name / engl.

Betriebssysteme / Operating Systems

Kürzel

6.2

Verantwortlicher

Dr. Volodymyr Brovko

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Sommersemester

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 5,

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

Lehrveranstaltungen

Betriebssysteme (3 SWS)

Praktikum Betriebssysteme (1 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Praktikum um den Stoff einzuüben.

Prüfung

Prüfungsnummer

-

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Schriftliche Prüfung, 60 Minuten, Hilfsmittel: 1 Blatt DIN-A4 handgeschriebene Formelsammlung

Inhalte des Moduls

- Grundstrukturen und Arbeitsweisen von Betriebssystemen
- Prozesse, Threads und Scheduling
- Synchronisation und Kommunikation
- Speicherverwaltung
- Ein-/Ausgabe
- Dateisysteme
- Sicherheit in Betriebssystemen

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Die Studierenden sind in der Lage,

- die Grundstrukturen von Betriebssystemen zu nennen und zu identifizieren.
- die wesentliche Arbeitsweise der Prozessverwaltung, der Speicherverwaltung sowie des Ein-/Ausgabesystems eines Betriebssystems zu erklären.
- den Aufbau und die Arbeitsweise eines Gerätetreibers zu erklären.
- POSIX-Systemfunktionen zu benutzen, um systemnahe Software zu implementieren.
- geläufige Synchronisationsmechanismen richtig anzuwenden, um parallele Anwendungen korrekt zu implementieren.
- die Effizienz von Software im Hinblick auf die Nutzung von Betriebssystem-Ressourcen zu analysieren und zu beurteilen.

Literaturliste

William Stallings: Operating Systems - Internals and Design Principles, 9. Auflage, Pearson, 2018, ISBN: 9780134700069

Andrew S. Tanenbaum, Herbert Bos: Modern Operating Systems, 4. Auflage, Pearson, 2015, ISBN: 978-1-292-06142-9, 1-292-06142-1

Jürgen Quade, Eva-Katharina Kunst: Linux-Treiber entwickeln, 4. Auflage, dpunkt.verlag, 2016

Christian Braun: Operating Systems / Betriebssysteme : Bilingual Edition: English – German / Zweisprachige Ausgabe: Englisch – Deutsch, Springer Vieweg, 2023, ISBN: 9783658422301

Robert Baumgartl: Betriebssysteme für dummies, Wiley Verlag, 2024, ISBN: 9783527718139

2.13 Seminar Neue Technologien

Name / engl.

Seminar Neue Technologien / Seminar New Technologies

Kürzel

3.4

Verantwortlicher

Professorinnen und Professoren der Fakultät für Informatik

Lehrsprache

Das Modul wird in englischer oder deutscher Sprache unterrichtet.

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Sommersemester

Arbeitsaufwand / Zusammensetzung

SWS: 2, CPs: 5,

Präsenzzeit: 30 h, Selbststudium: 120 h, Gesamtaufwand: 150 h

Lehrveranstaltungen

Seminar Neue Technologien (2 SWS)

Lehr-/Lernmethoden

Seminar

Prüfung

Prüfungsnummer

-

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Portfolioprüfung:

1. Praktische Prüfung, Arbeitsaufwand: max. 22h
 2. Studienarbeit, 10-15 Seiten, 60%
 3. Präsentation, 15-30 Minuten, 40%
-

Ergänzende Hinweise zur Prüfungsform

Die Note für das Modul wird wie folgt aus den Teilleistungen gebildet:

- Wird der erste Teil nicht bestanden, gilt das Modul als nicht bestanden.
 - Wird der erste Teil bestanden, wird die Modulnote aus den Noten des zweiten und dritten Teils mit folgender Gewichtung berechnet:
 - Studienarbeit: 60%
 - Präsentation: 40%
-

Inhalte des Moduls

Die Studierenden führen eine Recherche zu einem Thema aktuell relevanter Technologien im Bereich der Wirtschaftsinformatik durch. Sie dokumentieren, präsentieren und diskutieren im Plenum ihre Ergebnisse.

Zur Sicherstellung der Lernziele besteht in diesem Modul eine Anwesenheitspflicht.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Die Studierenden kennen die Grundsätze wissenschaftlicher Recherchen, wissenschaftlichen Schreibens und Präsentierens und sind in der Lage, Informationen zu fachspezifischen Themen

- zu sammeln.
- korrekt zu zitieren.
- mit eigenen Worten schriftlich und mündlich wiederzugeben.

Die Studierenden sind weiterhin in der Lage die Themen zu präsentieren und themenbezogen zu debattieren.

Literaturliste

Literatur wird in der Veranstaltung bekanntgegeben.

2.14 Fachwissenschaftliche Wahlpflichtfächer

Name / engl.

Fachwissenschaftliche Wahlpflichtfächer / Optional required Subjects

Kürzel W.1	Verantwortlicher Professorinnen und Professoren der Fakultät für Informatik
Lehrsprache Siehe Angaben des jeweiligen FWP-Fachs	Fakultät Fakultät für Informatik
Verwendbarkeit Informatik Bachelor	Dauer / Angebot Die Dauer des Moduls beträgt ein Semester. Das Modul wird sowohl im Sommersemester als auch im Wintersemester angeboten.
Arbeitsaufwand / Zusammensetzung CP: 30 Entnehmen Sie bitte die Zeiten für die Fächer dem Katalog für die FWP Fächer.	
Lehrveranstaltungen Die FWP-Fächer können aus dem Angebot der Fakultät für Informatik ausgewählt werden.	
Lehr-/Lernmethoden Seminaristischer Unterricht, Seminar, Praktikum, Directed Reading	

Prüfung

Prüfungsnummer	Benotung Gemäß § 20 der APO in der jeweils gültigen Fassung.
-----------------------	--

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

Zusätzliche Informationen

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

Inhalte des Moduls

Spezifische Fachkompetenz in den einzelnen Fächern.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Spezifische Fachkompetenz in den jeweiligen Fächern.

Gewichtung der Einzelleistung in der Gesamtnote

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

Literaturliste

Literatur wird im Seminar bekannt gegeben.

2.15 Allgemeinwissenschaftliches Wahlpflichtfach

Name / engl.

Allgemeinwissenschaftliches Wahlpflichtfach / General Science Elective Subjects

Kürzel

W.2

Verantwortlicher

Studiengangsleiter

Lehrsprache

Das Modul wird in deutscher und englischer Sprache unterrichtet.

Fakultät

Fakultät für angewandte Geistes- und Naturwissenschaften

Dauer / Angebot

Die Dauer des Moduls beträgt ein Semester.

Verwendbarkeit

Informatik Bachelor

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

Arbeitsaufwand / Zusammensetzung

CP: 2,
siehe Modulhandbuch für AWP-Fächer der Fakultät für angewandte Geistes- und Naturwissenschaften

Lehrveranstaltungen

Als allgemeinwissenschaftliche Wahlpflichtfächer können alle an der Hochschule Augsburg angebotenen Lehrveranstaltungen gewählt werden, soweit sie nicht Pflicht- oder Wahlpflichtfächer dieses Studiengangs sind bzw. in der Ausschlussliste des Studiengangs geführt werden.

Lehr-/Lernmethoden

Die Lehr- und Lernmethoden, sowie die verwendeten Lehrmedien variieren je nach Veranstaltung.

Prüfung

Prüfungsnummer

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Die Art und Dauer der Prüfung variiert je nach Veranstaltung und ist dem Modulhandbuch der Fakultät für angewandte Geistes- und Naturwissenschaften zu entnehmen.

Zusätzliche Informationen

hilfreiche Voraussetzungen

Informationen zur Teilnahme an Wahlpflichtfächern können unter §7 der Studien- und Prüfungsordnung entnommen werden.

Inhalte des Moduls

Die allgemeinwissenschaftliche Ausbildung an der Hochschule Augsburg umfasst ein vielseitiges Angebot in geistes-, gesellschafts- und naturwissenschaftlichen Fächern. Die Studierenden lernen Wissensgebiete kennen, die über ihr fachspezifisches Studium hinausgehen.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Die allgemeinwissenschaftlichen Wahlpflichtfächer stellen gewissermaßen ein "Studium generale" dar. Die Studierenden sind in der Lage, das erworbene theoretische Wissen in Studium und Beruf praktisch anzuwenden.

Gewichtung der Einzelleistung in der Gesamtnote

Die Art und Dauer der Prüfung variiert je nach Veranstaltung und ist dem Modulhandbuch der Fakultät für angewandte Geistes- und Naturwissenschaften zu entnehmen.

Literaturliste

Die Literaturliste ist dem Modulhandbuch der Fakultät für angewandte Geistes- und Naturwissenschaften zu entnehmen.

2.16 Praktische Tätigkeit - Praxissemester

Name / engl.

Praktische Tätigkeit - Praxissemester / **Practical Term**

Kürzel

P.1

Verantwortlicher

Praktikantenbeauftragte

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Wintersemester

Arbeitsaufwand / Zusammensetzung

ECTS: 20, Gesamtaufwand: 20 Wochen

Lehrveranstaltungen

Praktische Tätigkeit (20 Wochen)

Lehr-/Lernmethoden

Praktische Tätigkeit

formale Voraussetzungen

Die Aufnahme der praktischen Tätigkeit und die Teilnahme am Praxisseminar ist zulässig, wenn mindestens 80 ECTS nachgewiesen wurden.

Prüfung

Prüfungsnummer

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Praxisbericht, 20-50 Seiten

Inhalte des Moduls

Besonderheit:

Als Besonderheit des Studiums an bayerischen Hochschulen bieten wir Ihnen ein in das Studium integriertes, gesetzlich vorgeschriebenes praktisches Studiensemester, in welchem der Schwerpunkt der Wissensvermittlung in die Praxis hinaus verlegt wird. Während des Praxissemesters behalten Sie Ihren Status als Studentin oder Student bei, die praktische Ausbildung wird durch begleitende Unterrichtsveranstaltungen an der Hochschule ergänzt und vertieft.

Zuständig für die formale Abwicklung des Praktikums ist das Praktikantenamt. Lesen Sie deshalb bitte auch den Leitfaden für die praktischen Studiensemester des Praktikantenamtes.

Neben dem Praktikantenamt steht Ihnen ein fachlicher Betreuer zur Seite. Sprechen Sie ihn bitte insbesondere dann möglichst frühzeitig an, wenn es mit Ihrer Praktischen Tätigkeit irgendwelche Probleme gibt.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

- Anleitung zum selbstständigen und eigenverantwortlichen Arbeiten
- Einführung in das Berufsfeld durch möglichst selbstständige und eigenverantwortliche Mitarbeit
- Erweiterung und Vertiefung der Kenntnisse über organisatorische Problemlösungen im Betrieb
- Kenntnisse über Fragen der Berufsausübung wie Tätigkeitsmöglichkeiten, arbeitsrechtliche Formen und Arbeitsplätze
- Einblick in relevante Steuerverordnungen und soziale Absicherungen.

Gewichtung der Einzelleistung in der Gesamtnote

mit Erfolg abgelegt / ohne Erfolg abgelegt

Literaturliste

Falls notwendig wird die Literatur im Praktikumsbetrieb bekannt gegeben.

2.17 Praxisseminar

Name / engl.

Praxisseminar / Practical Term Mentoring Workshop

Kürzel P.2	Verantwortlicher Professorinnen und Professoren der Fakultät für Informatik
----------------------	---

Lehrsprache Deutsch	Fakultät Fakultät für Informatik
-------------------------------	--

Verwendbarkeit Informatik Bachelor	Dauer / Angebot ein Semester, jeweils im Wintersemester
--	---

Arbeitsaufwand / Zusammensetzung
SWS: 2, CPs: 3,
Präsenzzeit: 30 h, Selbststudium: 60 h, Gesamtaufwand: 90 h

Lehrveranstaltungen
Praxisseminar (2 SWS)

Lehr-/Lernmethoden
Seminar

formale Voraussetzungen
Die Aufnahme der praktischen Tätigkeit und die Teilnahme am Praxisseminar ist zulässig, wenn mindestens 80 ECTS nachgewiesen wurden.

Prüfung

Prüfungsnummer	Benotung Gemäß § 20 der APO in der jeweils gültigen Fassung.
-----------------------	--

Prüfungsform
Portfolioprüfung:
1. Präsentation, 15-30 Minuten
2. Praktische Prüfung, Arbeitsumfang: max. 22h
Beide Teile müssen bestanden werden.

Inhalte des Moduls

Studierende können

- die eigene Arbeit nach wissenschaftlichen Standards korrekt und nachvollziehbar präsentieren sowie Fragen beantworten.
- Präsentationen zu anderen Arbeiten verstehen und sich an fachlichen Diskussionen beteiligen.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Die Studierenden erweitern die Kompetenz für Präsentationen.

Gewichtung der Einzelleistung in der Gesamtnote

mit Erfolg abgelegt / ohne Erfolg abgelegt

Literaturliste

Literatur wird im Seminar bekannt gegeben.

2.18 Bachelorarbeit

Name / engl.

Bachelorarbeit / Bachelor Thesis

Kürzel

7.1

Verantwortlicher

Professorinnen und Professoren der Fakultät für Informatik

Lehrsprache

Das Modul wird in deutscher und/oder englischer Sprache unterrichtet.

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

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

Arbeitsaufwand / Zusammensetzung

ECTS: 12, Arbeitsstunden: 360 h

Bearbeitungszeit: 4 Monate

Lehrveranstaltungen

Bachelorarbeit

Lehr-/Lernmethoden

Wissenschaftliches Arbeiten

Prüfung

Prüfungsnummer

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Studienarbeit, 20-80 Seiten

Zusätzliche Informationen

hilfreiche Voraussetzungen

Informationen zur Bachelorarbeit können unter § 11 der Studien- und Prüfungsordnung entnommen werden.

Inhalte des Moduls

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Literaturliste

Fachliteratur zur gewählten Fragestellung.

2.19 Bachelorseminar

Name / engl.

Bachelorseminar / Scientific Working Seminar for Bachelor

Kürzel

7.2

Verantwortlicher

Professorinnen und Professoren der Fakultät für Informatik

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

Die Dauer des Moduls beträgt ein Semester.
Das Modul wird regelmäßig sowohl im Sommersemester als auch im Wintersemester angeboten.

Arbeitsaufwand / Zusammensetzung

SWS: 2, CPs: 2,

Präsenzzeit: 30 h, Selbststudium: 30 h, Gesamtaufwand: 60 h

Lehrveranstaltungen

Bachelorseminar: Wissenschaftliches Arbeiten (2 SWS)

Lehr-/Lernmethoden

Seminar, Coaching, Praktikum, Übung

formale Voraussetzungen

Das Bachelorseminar wird begleitend zur Bachelorarbeit durchgeführt. Die Anmeldung erfolgt automatisch mit der Anmeldung der Bachelorarbeit. (Um zu diesem Seminar zugelassen zu werden, muss der Teilnehmer zur Bachelorarbeit angemeldet sein.)

Prüfung

Prüfungsnummer

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Präsentation, 20 Minuten

Inhalte des Moduls

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Ziel ist es, die Studierenden begleitend zur Bachelorarbeit an eine geeignete wissenschaftliche Methodik heranzuführen. Der die Bachelorarbeit betreuende Dozent ist gleichzeitig auch der Dozent für das Bachelorseminar. Die Organisation und der Inhalt des Bachelorseminars wird durch den jeweiligen Dozenten selbst festgelegt. Inhaltlich könnten im Bachelorseminar sowohl wissenschaftliches Arbeiten als auch fachliche Themen aus dem Umfeld der Bachelorarbeiten abgehandelt werden.

Gewichtung der Einzelleistung in der Gesamtnote

mit Erfolg abgelegt / ohne Erfolg abgelegt

Literaturliste

Es wird empfohlen mit Antritt des Seminars die Angebote der Hochschulbibliothek insbesondere zur „Recherche“ und „Zitieren“ zu nutzen. Hierzu können Sie die aktuellen Seminartermine auf folgender Webseite prüfen:

Seminare - Recherchieren, Wissenschaftliches Arbeiten, Zitieren und Literatur-/Wissensverwaltung: all das können Sie in unseren Bibliotheksseminaren an der Hochschule Augsburg lernen

Index

- Algorithmen und Datenstrukturen , 28
- Allgemeinwissenschaftliches
 - Wahlpflichtfach , 76
- Angewandtes Software Engineering , 54
- Bachelorarbeit , 82
- Bachelorseminar , 84
- Betriebssysteme , 68
- Computernetzwerke , 48
- Daten- & Informationsanalyse , 44
- Datenbanksysteme , 50
- Einführung in das maschinelle Lernen , 62
- Einführung in die Programmierung , 3
- Einführung in die Theoretische Informatik , 10
- Entwicklungslabor , 6
- Fachwissenschaftliche Wahlpflichtfächer , 74
- Fortgeschrittene Programmier Techniken , 34
- Mathematik 1 , 8
- Mathematik 2 , 26
- Objektorientierte Programmierung , 18
- Praktische Tätigkeit - Praxissemester , 78
- Praxisseminar , 80
- Projektarbeit 1 , 60
- Projektarbeit 2 , 66
- Rechnerarchitektur , 32
- Rechnerstrukturen , 14
- Seminar Neue Technologien , 72
- Sichere Softwareentwicklung , 58
- Software Engineering , 22
- Softwaremodellierung , 38
- Statistik , 40