



Studienplan (Curriculum) für das

Bachelorstudium

Software & Information Engineering

an der Technischen Universität Wien

Gültig ab 1. Oktober 2014

Inhaltsverzeichnis

1. Grundlage und Geltungsbereich	3
2. Qualifikationsprofil	3
3. Dauer und Umfang	5
4. Zulassung zum Bachelorstudium	5
5. Aufbau des Studiums	5
6. Lehrveranstaltungen	14
7. Studieneingangs- und Orientierungsphase	14
8. Prüfungsordnung	15
9. Studierbarkeit und Mobilität	16
10. Bachelorarbeit	16
11. Akademischer Grad	17
12. Integriertes Qualitätsmanagement	17
13. Inkrafttreten	18
14. Übergangsbestimmungen	19
A. Modulbeschreibungen	19
B. Lehrveranstaltungstypen	84
C. Zusammenfassung aller verpflichtenden Voraussetzungen	85
D. Semestereinteilung der Lehrveranstaltungen	85
E. Semestereinteilung für schiefeinsteigende Studierende	87

1. Grundlage und Geltungsbereich

Der vorliegende Studienplan definiert und regelt das ingenieurwissenschaftliche Bachelorstudium *Software & Information Engineering* an der Technischen Universität Wien. Es basiert auf dem Universitätsgesetz 2002 BGBl. I Nr. 120/2002 (UG) und dem Satzungsteil *Studienrechtliche Bestimmungen* der Technischen Universität Wien in der jeweils geltenden Fassung. Die Struktur und Ausgestaltung des Studiums orientieren sich am folgenden Qualifikationsprofil.

2. Qualifikationsprofil

Das Bachelorstudium *Software & Information Engineering* vermittelt eine breite, wissenschaftlich und methodisch hochwertige, auf dauerhaftes Wissen ausgerichtete Grundausbildung, welche die Absolventinnen und Absolventen sowohl für eine Weiterqualifizierung im Rahmen eines facheinschlägigen Masterstudiums als auch für eine Beschäftigung in beispielsweise folgenden Tätigkeitsbereichen befähigt und international konkurrenzfähig macht:

- Entwicklung informationsverarbeitender Systeme sowohl als Experte im Team als auch in leitender Funktion
- Unterstützende Aufgaben in der Forschung

Software Engineering beschäftigt sich mit der Entwicklung von Software von der Analyse über das Design und die Implementierung bis hin zur Qualitätssicherung und Wartung. Information Engineering beschäftigt sich mit der Erzeugung, Sammlung, Verarbeitung, Verteilung und Präsentation von Information. Beide Gebiete basieren auf wissenschaftlichen Prinzipien und Methoden.

Aufgrund der beruflichen Anforderungen werden im Bachelorstudium *Software & Information Engineering* Qualifikationen hinsichtlich folgender Kategorien vermittelt.

Fachliche und methodische Kenntnisse Das Studium vermittelt fortgeschrittene Kenntnisse im Bereich der Informatik und ein kritisches Verständnis ihrer Theorien und Grundsätze. Aufbauend auf

- Mathematik, Statistik und theoretischer Informatik
- Algorithmen, Datenstrukturen und Programmierung
- Grundkenntnissen über den Aufbau von Computersystemen

werden die spezifischen Teilbereiche des Software- und Informationsengineering vermittelt:

- Softwareerstellung und -wartung

- Erhebung, Modellierung und Aufbereitung von Information
- Methoden der statistischen Datenanalyse
- Paradigmen und Konzepte von Programmiersprachen
- Datenbanken, wissensbasierte Systeme
- Betriebssysteme, Übersetzer und abstrakte Maschinen
- Verteilte Systeme, Internetcomputing
- Interaktion zwischen Menschen und Maschinen

Kognitive und praktische Fertigkeiten Durch die praktische und theoretische Auseinandersetzung mit aktuellen Technologien, Methoden und Werkzeugen (wie Programmiersprachen und Entwicklungsumgebungen) werden folgende Fertigkeiten vermittelt:

- Modellbildung und Abstraktion
- Einsatz formaler Grundlagen und Methoden zur Modellbildung, Lösungsfindung und Evaluation
- interdisziplinäre, systemorientierte und flexible Denkweise
- methodisch fundierte Herangehensweise an Probleme, insbesondere im Umgang mit unspezifizierten Problemsituationen
- kritische Bewertung und Reflexion von Lösungen
- hochwertige Dokumentation und überzeugende Präsentation

Soziale Kompetenzen, Innovationskompetenz und Kreativität Der Schwerpunkt liegt einerseits auf der Ausbildung berufsnotwendiger Zusatzkompetenzen, und andererseits auf der besonderen Förderung hoher Kreativitäts- und Innovationspotentiale.

- Selbstorganisation, Eigenverantwortlichkeit
- Teamfähigkeit im globalisierten Umfeld
- Eigeninitiative und Neugierde
- Finden kreativer Problemlösungen
- Entscheidungsverantwortung und Führungskompetenz in komplexen Projekten oder Tätigkeiten
- Kenntnisse der eigenen Fähigkeiten und Grenzen, Kritikfähigkeit
- Reflexion der eigenen Arbeit und ihrer Wechselwirkung mit dem gesellschaftlichen Kontext

3. Dauer und Umfang

Der Arbeitsaufwand für das Bachelorstudium *Software & Information Engineering* beträgt 180 ECTS-Punkte. Dies entspricht einer vorgesehenen Studiendauer von 6 Semestern als Vollzeitstudium.

ECTS-Punkte (Ects) sind ein Maß für den Arbeitsaufwand der Studierenden. Ein Studienjahr umfasst 60 ECTS-Punkte.

4. Zulassung zum Bachelorstudium

Voraussetzung für die Zulassung zum Bachelorstudium *Software & Information Engineering* ist die allgemeine Universitätsreife.

Personen, deren Muttersprache nicht Deutsch ist, haben die Kenntnis der deutschen Sprache nachzuweisen. Für einen erfolgreichen Studienfortgang werden Deutschkenntnisse nach Referenzniveau B2 des Gemeinsamen Europäischen Referenzrahmens für Sprachen empfohlen.

Lernunterlagen können in englischer Sprache abgefasst sein; weiters werden manche Lehrveranstaltungen auf Englisch angeboten. Daher werden Englischkenntnisse nach Referenzniveau B1 des Gemeinsamen Europäischen Referenzrahmens für Sprachen empfohlen.

5. Aufbau des Studiums

Die Inhalte und Qualifikationen des Studiums werden durch *Module* vermittelt. Ein Modul ist eine Lehr- und Lerneinheit, welche durch Eingangs- und Ausgangsqualifikationen, Inhalt, Lehr- und Lernformen, den Regelarbeitsaufwand sowie die Leistungsbeurteilung gekennzeichnet ist. Die Absolvierung von Modulen erfolgt in Form einzelner oder mehrerer inhaltlich zusammenhängender *Lehrveranstaltungen*. Thematisch ähnliche Module werden zu *Prüfungsfächern* zusammengefasst, deren Bezeichnung samt Umfang und Gesamtnote auf dem Abschlusszeugnis ausgewiesen wird.

Prüfungsfächer und zugehörige Module

Das Bachelorstudium *Software & Information Engineering* gliedert sich in nachstehende Prüfungsfächer mit den ihnen zugeordneten Modulen.

Die mit Stern markierten Module sind *Wahl-*, die übrigen *Pflichtmodule*. Die Pflichtmodule sind in jedem Fall zu absolvieren. Aus der Liste der Wahlmodule sind Module in einem Gesamtumfang von mindestens 24 Ects zu wählen. Im Rahmen des Moduls *Freie Wahl* sind so viele Lehrveranstaltungen zu absolvieren, dass ihr Umfang zusammen mit den 147 Ects der übrigen Pflichtmodule und dem Umfang der gewählten Wahlmodule mindestens 180 Ects ergibt.

Algorithmen und Programmierung

Algorithmen und Datenstrukturen (9.0 Ects)

Programmierparadigmen (6.0 Ects)

Programmkonstruktion (8.8 Ects)

Studieneingangsgespräch (0.2 Ects)

*Deklaratives Problemlösen (6.0 Ects)

*Logikprogrammierung und Constraints (6.0 Ects)

*Parallel Computing (6.0 Ects)

Computersysteme

Betriebssysteme (6.0 Ects)

Einführung in Visual Computing (6.0 Ects)

Technische Grundlagen der Informatik (6.0 Ects)

Verteilte Systeme (6.0 Ects)

*Abstrakte Maschinen (6.0 Ects)

*Microcontroller und Betriebssysteme (10.0 Ects)

*Übersetzerbau (6.0 Ects)

*Zuverlässige Echtzeitsysteme (6.0 Ects)

Informatik und Gesellschaft

Grundlagen der Human Computer Interaction (6.0 Ects)

Kontexte der Systementwicklung (6.0 Ects)

Security und Recht (6.0 Ects)

*Security (6.0 Ects)

*Vertrags-, Daten- und Informatikrecht (6.0 Ects)

Information Engineering

Datenbanksysteme (6.0 Ects)

Grundlagen intelligenter Systeme (8.0 Ects)

*Entwicklung von Web-Anwendungen (6.0 Ects)

*Modelle und Modellierung von statistischen Daten (7.5 Ects)

*Statistische Datenanalyse (6.0 Ects)

*Wissensrepräsentation (6.0 Ects)

Mathematik, Statistik und Theoretische Informatik

Algebra und Diskrete Mathematik (9.0 Ects)

Analysis (6.0 Ects)

Statistik und Wahrscheinlichkeitstheorie (6.0 Ects)

Theoretische Informatik und Logik (6.0 Ects)

- *Computernumerik (4.5 Ects)
- *Multivariate und computerintensive statistische Methoden (9.0 Ects)
- *Argumentieren und Beweisen (6.0 Ects)

Software Engineering

- Modellierung (9.0 Ects)
- Software Engineering und Projektmanagement (9.0 Ects)
- *Programm- und Systemverifikation (6.0 Ects)
- *Softwareprojekt-Beobachtung und -Controlling (6.0 Ects)
- *Softwarequalitätssicherung (6.0 Ects)
- *Usability Engineering and Mobile Interaction (6.0 Ects)

Fachübergreifende Qualifikationen und freie Wahl

- Fachübergreifende Qualifikationen (9.0 Ects)
- Freie Wahl (max. 9.0 Ects)

Bachelorarbeit

- Bachelorarbeit (13.0 Ects)

Kurzbeschreibung der Module

Dieser Abschnitt führt die Module des Bachelorstudiums *Software & Information Engineering* in alphabetischer Reihenfolge an und charakterisiert sie kurz. Eine ausführliche Beschreibung ist in Anhang A zu finden.

Abstrakte Maschinen (6.0 Ects) Dieses Modul vermittelt die theoretischen Grundlagen und konkrete Ausprägungen von abstrakten Maschinen. Dazu gehören Grundlagen über die effiziente Implementierung von abstrakten Maschinen und konkrete Maschinen wie die Java Virtual Machine, die Dalvik Virtual Machine, die Warren Abstract Machine und die SECD Maschine. Praktische Fertigkeiten werden durch die Implementierung einer eigenen abstrakten Maschine im Übungsteil vermittelt. Einfache Kenntnisse aus dem Übersetzerbau werden vorausgesetzt.

Algebra und Diskrete Mathematik (9.0 Ects) Das Modul bietet eine Einführung in die zentralen mathematische Grundlagen, Beweistechniken und Sätze in den Teilgebieten Algebra (v.a. algebraische Strukturen und lineare Algebra) und Diskrete Mathematik (v.a. Kombinatorik und Graphentheorie). Es setzt sich aus einem Vorlesungsteil und einem begleitenden Übungsteil zusammen, der der Vertiefung der Vorlesungsinhalte und der Entwicklung von Fertigkeiten zur Erstellung korrekter mathematischer Beweise und der mathematischen Modellierung und Analyse praktischer Problemstellungen dient.

Algorithmen und Datenstrukturen (9.0 Ects) Dieses Modul behandelt folgende Inhalte: Analyse von Algorithmen (asymptotisches Laufzeitverhalten, Omega, O- und

Theta-Notation); fundamentale Datentypen und Datenstrukturen; Sortieren und Suchen; grundlegende Graphenalgorithmien; Problemlösungsstrategien und Optimierung mit exakten, approximativen und heuristischen Verfahren; randomisierte Algorithmen; grundlegende geometrische Algorithmen.

Analysis (6.0 Ects) Das Modul bietet eine Einführung in die zentralen mathematischen Grundlagen, Beweistechniken im Teilgebiet Analysis (v.a. Folgen und Reihen, Differential- und Integralrechnung in einer Variablen). Es setzt sich aus einem Vorlesungsteil und einem begleitenden Übungsteil zusammen, der der Vertiefung der Vorlesungsinhalte und der Entwicklung von Fertigkeiten zur Erstellung korrekter mathematischer Beweise und der mathematischen Modellierung und Analyse praktischer Problemstellungen dient.

Argumentieren und Beweisen (6.0 Ects) Das Modul bietet eine Einführung in die zentralen Beweistechniken. Es setzt sich aus einem Vorlesungsteil und einem begleitenden Übungsteil zusammen, der der Vertiefung der Vorlesungsinhalte und der Entwicklung von Fertigkeiten zur Erstellung korrekter mathematischer Beweise dient. Schwerpunkte sind die Strukturierung von Beweisen und Argumentationen sowie die unterschiedlichen Techniken zur Induktion, die an praktischen Fragestellungen der Informatik demonstriert werden.

Bachelorarbeit (13.0 Ects) Ein Seminar führt in die wissenschaftliche Methodik und in den Wissenschaftsbetrieb ein. Darauf aufbauend bearbeitet der/die Studierende im Rahmen eines Projektes ein dem Qualifikationsprofil des Studiums entsprechendes Thema und beschreibt Aufgabenstellung, Methodik, Umfeld und Ergebnisse in einer schriftlichen Bachelorarbeit. Das Thema der Bachelorarbeit wird auf dem Abschlusszeugnis ausgewiesen.

Betriebssysteme (6.0 Ects) Dieses Modul vermittelt grundlegende Kenntnisse über Betriebssysteme, deren Architektur, Funktionsweise und wesentliche Komponenten. Die Grundkonzepte und theoretischen Inhalte werden einer Vorlesung, das Arbeiten mit Betriebssystemen und Betriebssystemmechanismen zusätzlich in praktischen Laborübungen vermittelt.

Vorausgesetzt werden Kenntnisse der Technischen Grundlagen der Informatik sowie Programmierkenntnisse.

Computernumerik (4.5 Ects) Studenten werden mit den grundlegenden Konzepten algorithmisch-numerischer Lösungsmethoden vertraut gemacht. Inhaltlich gehören dazu grundlegende Fehlerbegriffe, Kondition mathematischer Probleme, Datenfehler, Diskretisierungsfehler, Rundungsfehler, Numerische Lösung linearer Gleichungssysteme, numerische Differentiation und Integration, polynomiale Interpolation und Approximation, Design und Verwendung numerischer Algorithmen bzw. numerischer Software.

Datenbanksysteme (6.0 Ects) Dieses Modul vermittelt Grundkenntnisse von Datenbankmanagementsystemen, deren Architektur, wesentlichen Komponenten und Funktionsweise. Schwerpunkte liegen bei der Datenbank-Programmierung, physischer Datenorganisation und Anfragebearbeitung, Transaktionen, Fehlerbehandlung/Recovery, Mehrbenutzersynchronisation und verteilten Datenbanken.

Deklaratives Problemlösen (6.0 Ects) In diesem Modul werden vertiefende Kenntnisse zur Lösung komplexer computationaler Probleme mittels deklarativer Techniken vermittelt. Die Teilnehmer erlernen theoretische und anwendungsorientierte Aspekte unterschiedlicher Werkzeuge für deklaratives Problemlösen, welche auf klassischer Logik und Logikprogrammierung basieren.

Einführung in Visual Computing (6.0 Ects) Das Modul Einführung in Visual Computing vermittelt einen Überblick über die Aufgaben und Problemstellungen sowie die Methoden des Visual Computing, und ein kritisches Verständnis ihrer Theorien und Grundsätze. Der Begriff Visual Computing ist durch das methodische Zusammenwachsen der Bereiche Bildverarbeitung, Computer Vision, Computergraphik, Visualisierung und Mensch-Maschine-Interaktion entstanden, und umfasst außer diesen Themen auch Bereiche wie Augmented und Virtual Reality und maschinelles Lernen. Um dieses Modul absolvieren zu können werden Grundkenntnisse im Programmieren und solide Mathematikkenntnisse (Maturaniveau + Mathematik 1) vorausgesetzt.

Entwicklung von Web-Anwendungen (6.0 Ects) Das Modul *Entwicklung von Web-Anwendungen* beschäftigt sich einerseits mit der Aufbereitung und Verarbeitung von semistrukturierten Daten und andererseits mit Technologien und Entwicklungskonzepten zur Realisierung dynamischer Web-Anwendungen unter Berücksichtigung geltender Standards, u.a. für Barrierefreiheit.

Fachübergreifende Qualifikationen (9.0 Ects) Die Lehrveranstaltungen dieses Moduls dienen dem Erwerb fachübergreifender Qualifikationen wie zum Beispiel: Verhandlungsführung, Präsentations- und Kommunikationstechnik, systematische Recherche und Planung, Konfliktmanagement, Teamfähigkeit und Führung, Organisation und Management, Betriebsgründung und Finanzierung, Verständnis rechtlicher Rahmenbedingungen, Verbesserung von Fremdsprachenkenntnissen.

Freie Wahl (max. 9.0 Ects) Die Lehrveranstaltungen dieses Moduls dienen der Vertiefung des Faches sowie der Aneignung außerfachlicher Kenntnisse, Fähigkeiten und Kompetenzen.

Grundlagen der Human Computer Interaction (6.0 Ects) Dieses Modul vermittelt ein Verständnis dafür, wie sich Informationstechnologien sowie Designentscheidungen bei der Systementwicklung auf die Gesellschaft als Ganzes und auf Anwender im Einzelnen auswirken. Die Studierenden erlernen die grundlegenden Konzepte, Fähigkeiten und Prozesse für die Gestaltung von *Technik für Menschen* (Motto der TU Wien).

Grundlagen intelligenter Systeme (8.0 Ects) Studierende mit elementaren Logikkenntnissen, mit Kenntnissen in Datenstrukturen und Algorithmen und Fertigkeiten in der Mathematik (wie z.B. Beweise selbst zu führen) erhalten in diesem Modul (a) grundlegende Kenntnisse in den theoretischen Grundlagen intelligenter Systeme und (b) fundamentale Konzepte, die zum Verständnis der Arbeitsweise als auch zur Erstellung intelligenter Systeme notwendig sind. Das Modul deckt die Stoffgebiete Künstliche Intelligenz und Einführung in wissensbasierte Systeme ab. Beide Themengebiete werden in Vorlesungen mit zugehörigen Übungen vermittelt.

Kontexte der Systementwicklung (6.0 Ects) Dieses Modul vermittelt ein Verständnis für die Relevanz von gesellschaftlichem Umfeld, Ethik sowie Arbeits- und Freizeitgestaltung bei der Entwicklung von Software. Die Studierenden lernen die Anforderungen verstehen, die sich daraus für das Design von interaktiven Systemen, Abläufen und Mensch-Computer Schnittstellen ergeben.

Logikprogrammierung und Constraints (6.0 Ects) Dieses Modul vermittelt das logikorientierte Programmierparadigma anhand praktischer Aufgaben. Als Grundprogrammiersprache wird ISO-Prolog verwendet. Schwerpunkt ist der pure und monotone Teil der Sprache. Darauf aufbauend kommen in ISO-Prolog eingebettete Constraint-Programmiersprachen zum Einsatz.

Microcontroller und Betriebssysteme (10.0 Ects) Das Modul ist der Funktionsweise und Programmierung von Microcontroller-Systemen gewidmet. Ausgehend von der Vorstellung typischer Microcontroller-Architekturen und Peripheriekomponenten wird in begleitenden Laborübungen die gesamte Palette der Microcontroller-Programmierung abgedeckt, von der Assemblerprogrammierung über die Programmierung in einer höheren Programmiersprache bis zur Programmierung unter einem Microcontroller-Betriebssystem. Die komplementär dazu erfolgende praktische Beschäftigung mit dem Design, der Programmierung und Analyse von Betriebssystemen und deren Komponenten wird darüberhinaus Verständnis für jene komplexen Probleme und Aspekte geweckt, die unterhalb typischer Betriebssystem-Interfaces wie POSIX verborgen sind.

Modelle und Modellierung von statistischen Daten (7.5 Ects) Fuzzy Modelle behandeln Strategien zur Handhabung unscharfer Information, die in Form von Intervalldaten, verbaler Information, oder in noch allgemeinerer Form vorliegen können. Mit Computerstatistik wird ein Zugang zu statistischen Methoden vermittelt, der sowohl formal orientiert ist, bei dem aber auch anhand von konkreten Problemstellungen die theoretischen Konzepte praxisnah mit dem Computer gelöst werden.

Modellierung (9.0 Ects) Die Modellierung beschäftigt sich mit dem Prozess der Erstellung eines Modells als geeignete Abstraktion eines Realitätsausschnitts bzw. Systems. Der intendierte Verwendungszweck des Modells bestimmt, was als geeignete Abstraktion erachtet wird und welche Eigenschaften der Realität bzw. des Systems mit welchen Konzepten spezifiziert werden. Dieses Modul beschäftigt sich insbesondere mit den formalen Grundlagen der Modellbildung in der Informatik und dem Einsatz der Modellbildung für statische Systeme (Datenbanken) und objektorientierte Systeme.

Multivariate und computerintensive statistische Methoden (9.0 Ects) Dieses Modul vermittelt multivariate statistische Methoden und computerintensive Methodiken der Statistik mittels statistischer Simulation.

Parallel Computing (6.0 Ects) Der Kurs ist eine Einführung in das parallele Rechnen, und behandelt parallele Rechnerarchitekturen, Grundlagen der Parallelisierung, Elemente der Leistungsanalyse, parallele Programmiersprachen und -schnittstellen. Ein Schwerpunkt sind gängige Parallelisierungsschnittstellen, die für wissenschaftliches und Hochleistungsrechnen wichtig sind, wie MPI (Message-Passing Interface) und OpenMP.

Teilnehmer erwerben durch Projektübungen Kompetenz zur selbständigen Programmentwicklung mit diesen Schnittstellen. Schnittstellen und Sprachen für das Parallelprogrammieren von Mehrkernprozessoren werden ebenfalls berücksichtigt.

Programm- und Systemverifikation (6.0 Ects) Das Modul bietet eine Einführung in Methoden zur computerunterstützten Verifikation und Qualitätssicherung von Software und Hardware. Die in der Vorlesung vermittelten Grundlagen und Methoden werden an Hand von theoretischen und praktischen Aufgabenstellungen vertieft und in geeigneten Anwendungen erprobt.

Programmierparadigmen (6.0 Ects) Ein Programmierparadigma ist ein grundlegender Programmierstil. Aufbauend auf praktischen Programmierfertigkeiten sowie Kenntnissen der objektorientierten Modellierung und fundamentaler Algorithmen und Datenstrukturen gibt das Modul Programmierparadigmen einen vertieften Einblick in das objektorientierte und funktionale Paradigma und entsprechende Sprachkonstrukte, einschließlich solcher für Generizität, Nebenläufigkeit und Modularisierung. Studierende eignen sich durch die Beschäftigung mit typischen Problemstellungen bewährte Programmierstile an und lernen, die Stärken der Paradigmen zu nutzen, Schwierigkeiten zu umgehen, eigenständige Lösungsstrategien zu entwickeln und Lösungen kritisch zu bewerten.

Programmkonstruktion (8.8 Ects) Das Modul *Programmkonstruktion* führt Anfänger in die Programmierung ein, wobei der Schwerpunkt auf einer systematischen Vorgehensweise bei der Erstellung und Evaluierung von Programmen in einer objektorientierten Programmiersprache liegt. Neben Fachkenntnissen werden vor allem praktische Fertigkeiten in der Programmierung im Team (einschließlich des Einsatzes formaler und informeller Methoden) sowie abstrakte und systemorientierte Denkweisen vermittelt und die Neugierde an der Programmierung gefördert.

Security (6.0 Ects) IT-Sicherheit ist ein kritisches Element erfolgreicher IT-Projekte. Trotz funktional gut ausgeführter Projekte können diese bei schweren Sicherheitsproblemen je nach Anwendungsgebiet geschäftsschädigende Auswirkungen haben. In den Lehrveranstaltungen dieses Moduls lernen die Studierenden Sicherheitsprobleme zu erkennen und Sicherheitsmaßnahmen anzuwenden, um IT-Projekte auch aus Sicherheits-sicht erfolgreich abzuschließen.

Security und Recht (6.0 Ects) Dieses Modul eröffnet den Zugang zu den für das Internet bzw. die Informationsgesellschaft relevanten sicherheitsrelevanten und rechtlichen Aspekten und sensibilisiert für aktuelle sicherheitsrelevante und rechtspolitische Problemstellungen. Ferner leistet es einen Beitrag zur Reduktion der rechtlichen Risiken, denen Techniker/innen im Rahmen ihrer beruflichen Praxis ausgesetzt sind. Zusätzlich wird in diesem Modul das Aufgabengebiet der IT-Sicherheit aus verschiedenen Blickwinkeln präsentiert.

Software Engineering und Projektmanagement (9.0 Ects) Das Modul Software Engineering und Projektmanagement vermittelt den Studierenden grundlegende Kenntnisse zur Softwareerstellung und Wartung durch das Zusammenführen der isolierten

Kenntnisse und Fähigkeiten aus den relevanten vorangehenden Lehrveranstaltungen zu einer praxisnahen Gesamtsicht von der softwaretechnischen Problemstellung bis zur Lösung. Dazu gehören insbesondere Vorgehensmodelle und Rollen im Software Engineering, Anforderungsanalyse und Spezifikation, Systementwurf, Methoden der Implementierung, Integration und Test sowie Grundkenntnisse des Projektmanagements und Qualitätssicherung im Kontext der Softwareentwicklung. Das Modul setzt sich zusammen aus einer Vorlesung, in der die theoretischen Konzepte und methodischen Grundlagen vorgestellt sowie Erfahrungen aus der praktischen Übung reflektiert werden, und einer Übung, in der ein mittelgroßes Software Engineering Projekt mit dem Ziel eines real brauchbaren Software-Prototyps und zugehöriger Dokumentation in einer Kleingruppe mit intensiver Betreuung durchgeführt wird.

Softwareprojekt-Beobachtung und -Controlling (6.0 Ects) Das Vertiefungsmodul *Softwareprojekt-Beobachtung und -Controlling* vermittelt eine Sichtweise von IT- und Software-Projekten, die oberhalb des klassischen Projektmanagements in der Softwareentwicklung und unterhalb des sehr übergreifenden IT-Controllings als gesamtbetriebliche Instanz liegt. Auf Basis praxisnaher Beispiele werden Methoden zur Überwachung und Steuerung komplexer Projektsituationen präsentiert. Das Lehrziel sind Fertigkeiten aus folgenden Themengebieten:

- Identifikation von Kennziffern für das Benchmarking und Evaluieren eines Projektportfolios
- Nutzbringende Darstellung der Dimensionen Zeit, Kosten und Qualitätsmetriken der SW Herstellung
- Supervision: Überwachung und Eskalationsmechanismus
- Strategie: Analyse und Behebung von Fehlstellungen in der SW Erzeugung; SE Theorie vs. SE Praxis
- Vergleichbarkeit: Controlling über mehrere SW Projekte innerhalb eines Unternehmens

Softwarequalitätssicherung (6.0 Ects) Das Modul *Software-Qualitätssicherung* vermittelt eine Einführung in formale und angewandte Kenntnisse, Methoden und Kompetenzen zur Beurteilung und Verbesserung der Qualität von Softwaresystemen im wissenschaftlichen und industriellen Umfeld. Der Fokus liegt auf Reviews und Testen von Artefakten aus der Entwicklung von Softwaresystemen, die aus mehreren Komponenten bestehen. Die Lehrveranstaltung setzt sich zusammen aus einem Vorlesungsteil, in dem die theoretischen Konzepte und Lösungsansätze vorgestellt werden, und einem Übungsteil, in dem praktische Beispiele aus den Bereichen Reviews und Testen am Computer umgesetzt werden.

Statistik und Wahrscheinlichkeitstheorie (6.0 Ects) Dieses Modul vermittelt Grundkenntnisse der Wahrscheinlichkeitstheorie und Statistik.

Statistische Datenanalyse (6.0 Ects) Dieses Modul vermittelt Grundkenntnisse der statistischen Datenanalyse und der computerorientierten Statistik.

Studieneingangsgespräch (0.2 Ects) Vor oder zu Studienbeginn besprechen Lehrende mit den Studieninteressierten auf Basis eines Motivationsschreibens deren Interessen und Fähigkeiten, damit diese eine fundierte Entscheidungsgrundlage für oder gegen das geplante Studium erhalten.

Technische Grundlagen der Informatik (6.0 Ects) Das Modul vermittelt Kenntnisse im Bereich von Zahlendarstellungen, Boole'scher Algebra, Schaltnetzen und Schaltwerken, Grundlagen digitaler Systeme, Aufbau und Funktionsweise von Prozessoren und Computersystemen, Speicherverwaltung und Systemsoftware sowie peripheren Geräten.

Theoretische Informatik und Logik (6.0 Ects) Aufbauend auf elementaren Kenntnissen formaler Modellierungssprachen (wie Automaten oder Aussagenlogik) zur Spezifikation realer Sachverhalte vermittelt dieses Modul die theoretischen und logischen Grundlagen der Informatik und die Fähigkeit, formal-mathematische Beschreibungen verstehen und verfassen zu können.

Übersetzerbau (6.0 Ects) Das Modul vermittelt die theoretischen Grundlagen des Übersetzerbaus und die praktischen Fähigkeiten der Entwicklung von Parsern und Übersetzern. Es werden alle Phasen eines Übersetzers von der lexikalischen Analyse, der Syntaxanalyse, der semantischen Analyse, der Optimierung und der Codeerzeugung abgedeckt. Weiters wird noch auf die Implementierung von objektorientierten Programmiersprachen eingegangen. In Vorlesungen werden die theoretischen Grundlagen vermittelt, in einer Laborübung in geführten Kleingruppen werden die Inhalte in Form von Programmieraufgaben praktisch geübt.

Usability Engineering and Mobile Interaction (6.0 Ects) Dieses Modul vermittelt theoretische Grundlagen und praktische Methoden in den Bereichen Usability Engineering und User-Centered Interaction Research für mobile Anwendungen. Im Bereich Usability Engineering stehen die Qualitätskriterien für gute User Interfaces und die Methoden zu deren Evaluierung im Vordergrund. Aufbauend darauf wird im zweiten Teil des Moduls der Fokus auf den mobilen Bereich gelegt und dessen Besonderheiten anhand von Fallbeispielen hervorgehoben. Die in der Vorlesung vermittelten Kenntnisse sollen in den jeweiligen Übungsblöcken in Kleingruppen praktisch erprobt werden.

Verteilte Systeme (6.0 Ects) Das Modul *Verteilte Systeme* enthält folgende Lehrinhalte: Grundlagen und Konzepte, Middleware, Kommunikation, Operating System Support, Naming und Discovery, Synchronisation und Consensus, Replikation und Konsistenz, Fehlertoleranz, Dependability und Security, Technologieüberblick.

Vertrags-, Daten- und Informatikrecht (6.0 Ects) Dieses Verbreiterungsmodul soll zum einen die Teilnehmer/innen befähigen, konkrete Probleme des materiellen Internetrechts als solche zu erkennen, selbst zumindest grundsätzlich zu beurteilen sowie mit Jurist/inn/en bei der Lösung der Rechtsprobleme effektiv und kritisch auf interdisziplinärer Ebene zusammenzuarbeiten. Zum anderen sollen die Grundzüge und -mechanismen des

(privatrechtlichen) Vertrags- und Haftungsrechts präsentiert und das selbständige Lösen privatrechtlicher Problemstellungen trainiert werden.

Wissensrepräsentation (6.0 Ects) Aufbauend auf das Modul Grundlagen intelligenter Systeme vermittelt dieses Modul grundlegende Kenntnisse in Theorie und Anwendung der Wissensrepräsentation. Beginnend mit der klassischen Logik als Repräsentationssprache werden fortgeschrittenere Themen wie parakonsistentes Schließen, Wissensrevision (belief revision) und logikbasierte Abduktion behandelt. Es werden sowohl Algorithmen für diese Konzepte thematisiert als auch Einbettungsansätze (in andere Formalismen, für die gute Werkzeuge existieren) besprochen. Ein solcher Formalismus ist die quantifizierte Aussagenlogik, auf die auf Grund ihrer zentralen Bedeutung für mehrere Gebiete der Informatik detailliert eingegangen wird.

Zuverlässige Echtzeitsysteme (6.0 Ects) Das Modul vermittelt die wesentlichen Kenntnisse für die Spezifikation, den Entwurf, die Implementierung und das Testen von fehlertoleranten, sowie sicherheitskritischen verteilten Echtzeitsystemen. Fehlerarten, Fehlermodellierung, Fehlermaskierung, der Umgang mit zeitabhängiger Information, die Konstruktion von Computersystemen mit strikten Anforderungen im Zeitbereich und die Auswirkungen dieser Faktoren auf die Sicherheit von Computersystemen sind dabei zentrale Aspekte. Die Grundlagen zu zuverlässigen Systemen und Echtzeitsystemen werden in Vorlesungen vermittelt. Problemstellungen aus der Simulation von fehlertoleranten Systemen und der Fehleranalyse/modellierung werden in praktischen Übungen behandelt.

6. Lehrveranstaltungen

Die Stoffgebiete der Module werden durch Lehrveranstaltungen vermittelt. Die Lehrveranstaltungen der einzelnen Module sind in Anhang A in den jeweiligen Modulbeschreibungen spezifiziert. Lehrveranstaltungen werden durch Prüfungen im Sinne des Universitätsgesetzes beurteilt. Die Arten der Lehrveranstaltungsbeurteilungen sind in der Prüfungsordnung (siehe Abschnitt 8) festgelegt.

Änderungen an den Lehrveranstaltungen eines Moduls werden in der Evidenz der Module dokumentiert, mit Übergangsbestimmungen versehen und im Mitteilungsblatt der Technischen Universität Wien veröffentlicht. Die aktuell gültige Evidenz der Module liegt im Dekanat der Fakultät für Informatik auf.

7. Studieneingangs- und Orientierungsphase

Die Studieneingangs- und Orientierungsphase soll den Studierenden eine verlässliche Überprüfung ihrer Studienwahl ermöglichen. Sie leitet vom schulischen Lernen zum universitären Wissenserwerb über und schafft das Bewusstsein für die erforderliche Begabung und die nötige Leistungsbereitschaft.

Die Studieneingangs- und Orientierungsphase des Bachelorstudiums *Software & Information Engineering* umfasst folgende Module bzw. Lehrveranstaltungen:

Algebra und Diskrete Mathematik (9.0 Ects)
Modellierung (9.0 Ects)
Programmkonstruktion (8.8 Ects)
Studieneingangsgespräch (0.2 Ects)
Technische Grundlagen der Informatik (6.0 Ects)

wobei im Modul *Modellierung* nur die Lehrveranstaltungen *Formale Modellierung* (3.0 Ects) und *Datenmodellierung* (3.0 Ects) Teil der Studieneingangs- und Orientierungsphase sind.

Die positiv absolvierte Studieneingangs- und Orientierungsphase ist Voraussetzung für die Absolvierung aller Lehrveranstaltungen aus Modulen dieses Studienplans (inklusive der Bachelorarbeit) ausgenommen die Lehrveranstaltungen der Module

Algebra und Diskrete Mathematik (9.0 Ects)
Algorithmen und Datenstrukturen (9.0 Ects)
Analysis (6.0 Ects)
Einführung in Visual Computing (6.0 Ects)
Grundlagen der Human Computer Interaction (6.0 Ects)
Modellierung (9.0 Ects)
Programmkonstruktion (8.8 Ects)
Studieneingangsgespräch (0.2 Ects)
Technische Grundlagen der Informatik (6.0 Ects)

Studieneingangsgespräch

Vor oder zu Studienbeginn ist ein eigenständig verfasstes Motivationsschreiben abzugeben und in einem Studieneingangsgespräch mit Angehörigen der Fakultät zu besprechen, um die Gründe für die Studienwahl und die Erwartungen an das Studium zu reflektieren.

Das Studieneingangsgespräch ist durch das Modul *Studieneingangsgespräch* im Studienplan verankert. Die Absolvierung dieses Moduls – durch Abgabe des Motivations-schreibens und aktive Teilnahme am Gespräch – bildet die Voraussetzung für alle anderen Module des Studiums.

8. Prüfungsordnung

Für den Abschluss des Bachelorstudiums ist die positive Absolvierung der vom Studienplan vorgeschriebenen Module erforderlich. Ein Modul gilt als positiv absolviert, wenn die ihm zuzurechnenden Lehrveranstaltungen gemäß Modulbeschreibung positiv absolviert wurden.

Das Abschlusszeugnis beinhaltet

- (a) die Prüfungsfächer mit ihrem jeweiligen Umfang in ECTS-Punkten und ihren Noten,
- (b) das Thema der Bachelorarbeit,
- (c) die Gesamtbeurteilung gemäß UG § 73/3 sowie die Gesamtnote.

Die Note eines Prüfungsfaches ergibt sich durch Mittelung der Noten jener Lehrveranstaltungen, die dem Prüfungsfach über die darin enthaltenen Module zuzuordnen sind, wobei die Noten mit dem ECTS-Umfang der Lehrveranstaltungen gewichtet werden. Bei einem Nachkommateil kleiner gleich 0,5 wird abgerundet, andernfalls wird aufgerundet. Die Gesamtnote ergibt sich analog den Prüfungsfachnoten durch gewichtete Mittelung der Noten aller dem Studium zuzuordnenden Lehrveranstaltungen.

Die Studieneingangs- und Orientierungsphase gilt als positiv absolviert, wenn alle ihr zugeordneten Lehrveranstaltungen positiv absolviert wurden.

Lehrveranstaltungen des Typs VO (Vorlesung) werden aufgrund einer abschließenden mündlichen und/oder schriftlichen Prüfung beurteilt. Alle anderen Lehrveranstaltungen besitzen immanenten Prüfungscharakter, d.h., die Beurteilung erfolgt laufend durch eine begleitende Erfolgskontrolle sowie optional durch eine zusätzliche abschließende Teilprüfung.

Der positive Erfolg von Prüfungen ist mit „sehr gut“ (1), „gut“ (2), „befriedigend“ (3) oder „genügend“ (4), der negative Erfolg ist mit „nicht genügend“ (5) zu beurteilen. Die Beurteilung der Lehrveranstaltung

0.2/1.0 UE Studieneingangsgespräch erfolgt durch „mit Erfolg teilgenommen“ bzw. „ohne Erfolg teilgenommen“; sie bleibt bei der Berechnung der gemittelten Note des Prüfungsfaches unberücksichtigt.

9. Studierbarkeit und Mobilität

Studierende des Bachelorstudiums *Software & Information Engineering*, die ihre Studienwahl im Bewusstsein der erforderlichen Begabungen und der nötigen Leistungsbereitschaft getroffen und die Studieneingangs- und Orientierungsphase, die dieses Bewusstsein vermittelt, absolviert haben, sollen ihr Studium mit angemessenem Aufwand in der dafür vorgesehenen Zeit abschließen können.

Es wird empfohlen, das Studium nach dem Semestervorschlag in Anhang D zu absolvieren. Für Studierende, die ihr Studium im Sommersemester beginnen, wird der modifizierte Semestervorschlag in Anhang E empfohlen.

Die Anerkennung von im Ausland absolvierten Studienleistungen erfolgt durch das studienrechtliche Organ. Zur Erleichterung der Mobilität stehen die in § 27 Abs. 1 bis 3 der *Studienrechtlichen Bestimmungen* der Satzung der Technischen Universität Wien angeführten Möglichkeiten zur Verfügung. Diese Bestimmungen können in Einzelfällen auch zur Verbesserung der Studierbarkeit eingesetzt werden.

10. Bachelorarbeit

Die Bachelorarbeit ist eine im Bachelorstudium eigens angefertigte schriftliche Arbeit mit einem Regelarbeitsaufwand von 10 ECTS-Punkten, welche eigenständige Leistungen beinhaltet. Sie wird im Rahmen des Moduls *Bachelorarbeit* erstellt.

11. Akademischer Grad

Den Absolventinnen und Absolventen des Bachelorstudiums *Software & Information Engineering* wird der akademische Grad *Bachelor of Science* – abgekürzt *BSc* – verliehen.

12. Integriertes Qualitätsmanagement

Das integrierte Qualitätsmanagement gewährleistet, dass der Studienplan des Bachelorstudiums *Software & Information Engineering* konsistent konzipiert ist, effizient abgewickelt und regelmäßig überprüft bzw. kontrolliert wird. Geeignete Maßnahmen stellen die Relevanz und Aktualität des Studienplans sowie der einzelnen Lehrveranstaltungen im Zeitablauf sicher; für deren Festlegung und Überwachung sind das Studienrechtliche Organ und die Studienkommission zuständig.

Die semesterweise Lehrveranstaltungsbewertung liefert, ebenso wie individuelle Rückmeldungen zum Studienbetrieb an das Studienrechtliche Organ, für zumindest die Pflichtlehrveranstaltungen ein Gesamtbild für alle Beteiligten über die Abwicklung des Studienplans. Insbesondere können somit kritische Lehrveranstaltungen identifiziert und in Abstimmung zwischen studienrechtlichem Organ, Studienkommission und Lehrveranstaltungsleiterin und -leiter geeignete Anpassungsmaßnahmen abgeleitet und umgesetzt werden.

Die Studienkommission unterzieht den Studienplan in einem dreijährigen Zyklus einem Monitoring, unter Einbeziehung wissenschaftlicher Aspekte, Berücksichtigung externer Faktoren und Überprüfung der Arbeitsaufwände, um Verbesserungspotentiale des Studienplans zu identifizieren und die Aktualität zu gewährleisten.

Jedes Modul besitzt eine Modulverantwortliche oder einen Modulverantwortlichen. Diese Person ist für die inhaltliche Kohärenz und die Qualität der dem Modul zugeordneten Lehrveranstaltungen verantwortlich. Diese wird insbesondere durch zyklische Kontrollen, inhaltliche Feinabstimmung mit vorausgehenden und nachfolgenden Modulen sowie durch Vergleich mit analogen Lehrveranstaltungen bzw. Modulen anderer Universitäten im In- und Ausland sichergestellt.

Lehrveranstaltungskapazitäten

Für die verschiedenen Typen von Lehrveranstaltungen (siehe Anhang B) dienen die folgenden Gruppengrößen als Richtwert:

Lehrveranstaltungstyp	Gruppengröße	
	je Leiter(in)	je Tutor(in)
VO	200	
UE mit Tutor(inn)en	50	20
UE	20	
LU mit Tutor(inn)en	40	15
LU	15	
EX, PR, SE	20	

Für Lehrveranstaltungen des Typs VU werden für den Vorlesungs- bzw. Übungsteil die Gruppengrößen für VO bzw. UE herangezogen. Die Beauftragung der Lehrenden erfolgt entsprechend der tatsächlichen Abhaltung.

Lehrveranstaltungen mit ressourcenbedingten Teilnahmebeschränkungen sind in der Beschreibung des jeweiligen Moduls entsprechend gekennzeichnet; weiters sind dort die Anzahl der verfügbaren Plätze und das Verfahren zur Vergabe dieser Plätze festgelegt. Die Lehrveranstaltungsleiterinnen und Lehrveranstaltungsleiter sind berechtigt, mehr Teilnehmerinnen und Teilnehmer zu einer Lehrveranstaltung zulassen als nach Teilnahmebeschränkungen oder Gruppengrößen vorgesehen, sofern dadurch die Qualität der Lehre nicht beeinträchtigt wird.

Kommt es in einer Lehrveranstaltung ohne explizit geregelte Platzvergabe zu einem unvorhergesehenen Andrang, kann die Lehrveranstaltungsleitung in Absprache mit dem studienrechtlichen Organ Teilnahmebeschränkungen vornehmen und die Vergabe der Plätze nach folgenden Kriterien (mit absteigender Priorität) regeln.

- Es werden jene Studierenden bevorzugt aufgenommen, die die formalen und inhaltlichen Voraussetzungen erfüllen. Die inhaltlichen Voraussetzungen können etwa an Hand von bereits abgelegten Prüfungen oder durch einen Eingangstest überprüft werden.
- Unter diesen hat die Verwendung der Lehrveranstaltung als Pflichtfach Vorrang vor der Verwendung als Wahlfach und diese vor der Verwendung als Freifach.
- Innerhalb dieser drei Gruppen sind jeweils jene Studierenden zu bevorzugen, die trotz Vorliegens aller Voraussetzungen bereits in einem früheren Abhaltesemester abgewiesen wurden.

Die Studierenden sind darüber ehebaldigst zu informieren.

13. Inkrafttreten

Dieser Studienplan tritt mit 1. Oktober 2014 in Kraft.

14. Übergangsbestimmungen

Die Übergangsbestimmungen werden gesondert im Mitteilungsblatt verlautbart und liegen im Dekanat der Fakultät für Informatik auf.

A. Modulbeschreibungen

Abstrakte Maschinen

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse: Das Modul vermittelt

- alle theoretischen Grundlagen von abstrakten Maschinen und
- Kenntnisse über konkrete abstrakte Maschinen.

Kognitive und praktische Fertigkeiten: Die Auseinandersetzung mit konkreten Beispielen von abstrakten Maschinen und die Implementierung eigener abstrakter Maschinen vermittelt

- die Fähigkeit die Qualität von abstrakten Maschinen zu beurteilen,
- die praktische Fähigkeit zum Entwurf eigener abstrakter Maschinen und
- die praktische Fähigkeit zur Implementierung abstrakter Maschinen.

Soziale Kompetenzen, Innovationskompetenz und Kreativität: Eigeninitiative und Neugierde auf innovative und kreative Konzepte und Lösungsansätze werden besonders gefördert.

Inhalt:

- reale Maschinen, Prozessorarchitekturen
- Interpretationstechniken (threaded code), Implementierung von Forth
- Pascal P4 Maschine
- Java Virtuelle Maschine (just-in-time Übersetzung), Microsoft Intermediate Language
- Registermaschinen und die DalvikVM
- syntaxgesteuerte Editoren und Baummaschinen
- Prologmaschinen (WAM, VAM)

- funktionale Maschinen (Lamda Kalkül, SECD Maschine)

Erwartete Vorkenntnisse:

Fachliche und methodische Kenntnisse: Grundlagen von Programmiersprachen und Übersetzerbau

Kognitive und praktische Fertigkeiten: Programmierkenntnisse

Diese Voraussetzungen werden in folgenden Modulen vermittelt: *Programmkonstruktion, Programmierparadigmen, Übersetzerbau.*

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung:

- 3 ECTS Vortrag und selbständiges Erlernen der eher theoretischen Grundlagen. Die Beurteilung erfolgt durch Prüfung.
- 3 ECTS Übung am Computer zur Entwicklung praktischer Fähigkeiten zur Entwicklung von abstrakten Maschinen. Die Leistungsbeurteilung erfolgt durch die Beurteilung der Lösungen der Programmieraufgaben plus Abschlussgespräch.

Lehrveranstaltungen des Moduls:

3.0/2.0 VO Abstrakte Maschinen

3.0/2.0 UE Abstrakte Maschinen

Algebra und Diskrete Mathematik

Regelarbeitsaufwand: 9.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse: Vertrautheit mit den wichtigsten mathematischen Konzepten und Grundlagen in den Teilgebieten Algebra und Diskrete Mathematik.

Kognitive und praktische Fertigkeiten: Vertieftes Verständnis mathematischer Schlussweisen und Beweistechniken, Fertigkeit zur Erstellung mathematischer Beweise für einfache mathematische Probleme.

Soziale Kompetenzen, Innovationskompetenz und Kreativität: Mathematische Formulierung praktischer Problemstellungen aus Informatik, Naturwissenschaften und Technik und Verwendung geeigneter mathematischer Lösungsverfahren zur analytischen und numerischen Problemlösung.

Inhalt:

Grundlagen

- Elementare Logik (Aussagen, Implikation, Kontraposition, Verneinung, Quantoren)

- Elementare Beweistechniken (direkter und indirekter Beweis, Gegenbeispiele)
- Elementare Zahlentheorie

Mengenlehre

- Grundlagen (Venn-Diagramme, Komplemente, kartesisches Produkt, Potenzmenge)
- Funktionen (Mengenrelationen, surjektive, injektive, bijektive Funktionen, Komposition)
- Relationen (Äquivalenzrelation, Partitionen, Ordnungsrelation, Maximumsprinzip)
- Kardinalität und Abzählbarkeit (endliche, unendlichen und abzählbare Mengen)

Induktion

- Induktionsprinzip (vollständige Ind., transfinite Ind.)
- Rekursive Definitionen

Grundlagen Kombinatorik

- Abzählprinzipien (Summen- und Produktregel)
- Schubfachschluss
- Inklusions-Exklusions-Prinzip
- Kombinatorische Grundaufgaben (Permutationen, Auswahlen, Partitionen)
- Elementare Identitäten (Binomischer Lehrsatz, binomische Identitäten)
- Rekursionen (Fibonacci-Zahlen, Derangements, Turm von Hanoi, Catalan-Zahlen)
- Lösungsmethoden für Rekursionen (Rekursionen erster Ordnungen, lineare Rekursionen mit konstanten Koeffizienten)

Graphentheorie

- Grundlagen (gerichtete, ungerichtete, bipartite Graphen, Wege, etc.)
- Handshake-Lemma
- Eulersche und Hamiltonsche Linien
- Graphrelationen (Isomorphie, Subgraphen, Minore)
- Zusammenhang (Zusammenhangskomponenten, Menger's theorem)

- Azyklische Graphen
- Ebene Graphen (inkl. Eulersche Polyederformel)
- Elementare Graph-Algorithmen (Azyklizität, Kruskal-Alg., Minimaler Spannbaum, Dijkstra-Alg.)

Algebraische Strukturen

- Gruppentheorie (inkl. Faktorgruppen, Homomorphiesatz, zyklische Gruppen, direkte Produkte)
- Ringe (Integritätsbereiche, Ideale)
- Körper (Polynomringe über Körpern)
- Verbände

Lineare Algebra

- Vektoren
- Matrizen (inklusive Tensor-Produkt)
- Lineare Abbildungen
- Lineare Gleichungssysteme
- Determinanten
- Eigenwerte und Eigenvektoren
- Skalarprodukte, Orthogonalität

Grundlagen Algebraische Codierungstheorie

- Gruppencodes
- Linearcodes

Erwartete Vorkenntnisse: Fundierte Mathematik-Kenntnisse auf AHS/BHS-Maturaniveau.

Verpflichtende Voraussetzungen: Keine.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Wöchentliche 4-stündige Vorlesung mit kontinuierlicher begleitender 2-stündiger Übung (individuell auszuarbeitende Übungsbeispiele), wodurch die in der Vorlesung vermittelten Inhalte effizient erlernt und die mathematische Problemlösungskompetenz trainiert wird. Leistungsfeststellung durch mehrere Lösungsdemonstrationen, Übungstests, Abschlussprüfung.

Lehrveranstaltungen des Moduls:

- 4.0/4.0 VO Algebra und Diskrete Mathematik für Informatik und Wirtschaftsinformatik
- 5.0/2.0 UE Algebra und Diskrete Mathematik für Informatik und Wirtschaftsinformatik

Algorithmen und Datenstrukturen

Regelarbeitsaufwand: 9.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse: Das Modul vermittelt

- Methoden zur Bewertung und Analyse von Algorithmen,
- fundamentale Algorithmen und Datenstrukturen,
- effiziente Lösungsansätze für häufige Problemstellungen in der Programmentwicklung,
- und Kenntnisse über eine systematische Vorgehensweise zur Entwicklung neuer Algorithmen.

Kognitive und praktische Fertigkeiten: Die Auseinandersetzung mit den Inhalten dieses Moduls vermittelt

- eine abstrakte und effizienzorientierte Denkweise für die Entwicklung von Programmen,
- die Fähigkeit zum Einsatz formaler und informeller Methoden zur Analyse von Algorithmen
- sowie Kenntnisse zur adäquaten Anwendung fundamentaler Algorithmen und Datenstrukturen.

Soziale Kompetenzen, Innovationskompetenz und Kreativität: Folgende Kompetenzen werden besonders gefördert:

- Selbstorganisation und Eigenverantwortlichkeit,
- Neugierde an Entwicklung effizienter Algorithmen.

Inhalt:

- Analyse von Algorithmen, insbesondere Untersuchung von Laufzeit- und Speicherplatzverhalten
- Sortierprobleme und Sortierverfahren
- Suchprobleme und Suchverfahren
- Graphen
- Problemlösungsstrategien und Optimierung

- Suchen in Texten, Pattern Matching (Algorithmen von Knuth-Morris-Pratt und Boyer Moore, Tries)
- Randomisierte Algorithmen
- Grundlegende geometrische Algorithmen (Scan-Line Prinzip, mehrdimensionale Bereichssuche)

Erwartete Vorkenntnisse:

Fachliche und methodische Kenntnisse: Grundkenntnisse der linearen Algebra und Analysis, insbesondere Mengenlehre, Metriken, Folgen und Reihen.

Kognitive und praktische Fertigkeiten: Programmierkenntnisse

Diese Voraussetzungen werden in folgenden Modulen vermittelt: *Algebra und Diskrete Mathematik, Programmkonstruktion.*

Verpflichtende Voraussetzungen: Keine.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die Inhalte des Moduls werden im Rahmen einer Vorlesung (6 ECTS) präsentiert und an Hand ausgewählter Beispiele illustriert. Durch die Ausarbeitung von Aufgaben und deren Diskussion in Kleingruppen bei regelmäßigen Treffen (Anwesenheitspflicht!) vertiefen die Studierenden ihr Verständnis für den Stoff; in zusätzlichen Programmieraufgaben wird ferner die Umsetzung algorithmischer Aufgabenstellungen in der Praxis geübt (3 ECTS).

Lehrveranstaltungen des Moduls:

6.0/4.0 VU Algorithmen und Datenstrukturen 1

3.0/2.0 VU Algorithmen und Datenstrukturen 2

Analysis

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse: Vertrautheit mit den wichtigsten mathematischen Konzepten und Grundlagen im Teilgebiet Analysis.

Kognitive und praktische Fertigkeiten: Vertieftes Verständnis mathematischer Schlussweisen und Beweistechniken, Fertigkeit zur Erstellung mathematischer Beweise für einfache mathematische Probleme.

Soziale Kompetenzen, Innovationskompetenz und Kreativität: Mathematische Formulierung praktischer Problemstellungen aus Informatik, Naturwissenschaften und Technik und Verwendung geeigneter mathematischer Lösungsverfahren zur analytischen und numerischen Problemlösung.

Inhalt:

Folgen, Reihen und Funktionen

- Folgen reeller Zahlen (Grenzwert, Monotonie und Beschränktheit, Konvergenzuntersuchungen)
- Unendliche Reihen (Konvergenzkriterien, Cauchyprodukt und Potenzreihen)
- Asymptotischer Vergleich von Folgen (Landausymbole: $O()$, $o()$, $\Omega()$)

Elementare Funktionen

- Potenzen mit reellen Exponenten
- Exponentialfunktion und Logarithmus
- Darstellung der Exponentialfunktion
- Winkelfunktionen und Arcusfunktionen

Grenzwerte und Nullstellen von Funktionen, Stetigkeit

- Metrische und topologische Grundbegriffe (offene, geschlossene Mengen, Umgebungen, Basis, Häufungspunkte)
- Umgebungs- und Folgenstetigkeit
- Eigenschaften stetiger Funktionen: Nullstellensatz, Zwischenwertsatz, Monotonie
- Fixpunktsatz, Lipschitzbedingung
- Newton'sches Näherungsverfahren
- Die regula falsi

Differentialrechnung in einer Variablen

- Differenzenquotient und Differenzierbarkeit
- Ableitung einfacher Funktionen
- Eigenschaften und Ableitungsregeln
- Mittelwertsatz der Differentialrechnung
- Taylorreihen
- Monotonie und die erste Ableitung
- Höhere Ableitungen
- Der verallgemeinerte Mittelwertsatz und die Regel von de l'Hospital

Integralrechnung in einer Variablen

- Definition und Eigenschaften Riemann-Integral
- Integration als Umkehrung der Differentiation, Fläche unter Kurven
- Techniken des Integrierens
- Mittelwert- und Hauptsatz der Differential- und Integralrechnung
- Uneigentliche Integrale

Grundlagen Differential- und Integralrechnung in mehreren Variablen

- Funktionen in mehreren Variablen
- Partielle Ableitungen, totale Ableitung
- Ableitungsregeln
- Richtungsableitung
- Taylorentwicklung
- Hauptsatz über Implizite Funktionen
- Lokale Extrema
- Grundlagen Gebietsintegrale

Elementare Differentialgleichungen

- Lineare Differentialgleichungen erster Ordnung
- Lineare Differentialgleichungen höherer Ordnung mit konstanten Koeffizienten

Erwartete Vorkenntnisse: Fundierte Mathematik-Kenntnisse auf AHS/BHS-Maturaniveau.

Verpflichtende Voraussetzungen: Keine.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Wöchentliche 2-stündige Vorlesung mit kontinuierlicher begleitender 2-stündiger Übung (individuell auszuarbeitende Übungsbeispiele), wodurch die in der Vorlesung vermittelten Inhalte effizient erlernt und die mathematische Problemlösungskompetenz trainiert wird. Leistungsfeststellung durch mehrere Lösungsdemonstrationen, Übungstests, Abschlussprüfung.

Lehrveranstaltungen des Moduls:

- 2.0/2.0 VO Analysis für Informatik und Wirtschaftsinformatik
- 4.0/2.0 UE Analysis für Informatik und Wirtschaftsinformatik

Argumentieren und Beweisen

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse: Vertrautheit mit den wesentlichen mathematischen Schlussweisen und Beweistechniken.

Kognitive und praktische Fertigkeiten: Vertiefte Kenntnisse in Methoden zur Erstellung und Strukturierung von Beweisen, Fertigkeit zur Erstellung auch komplexer Beweise.

Soziale Kompetenzen, Innovationskompetenz und Kreativität: Kommunikation der Beweisideen und Beweise, Erstellen und Strukturieren von Beweisen für diverse Problemstellungen, kreative Erstellung von Hypothesen bei Induktionsbeweisen.

Inhalt:

- Was ist ein Beweis? Welche Aufgaben hat er?
- Einfache Beweistechniken
- Beweis von All- und Existenzaussagen, Konjunktionen, Disjunktionen, Implikationen, Äquivalenzen
- Nutzung dieser Aussagen in einem Beweis
- Zusammenhang zum Kalkül des natürlichen Schliessens
- Was ist Induktion? Wozu wird sie benötigt?
- Arten der Induktion (mathematische, starke, strukturelle, Noether'sche), jeweils mit Diskussion des entsprechenden Induktionsschemas und Anwendungsfälle (ausführlich demonstriert an Beispielen)
- Wie schreibt man einen Induktionsbeweis?

Erwartete Vorkenntnisse:

Fachliche und methodische Kenntnisse: Mathematikkenntnisse aus Algebra und Diskrete Mathematik

Kognitive und praktische Fertigkeiten: erste Erfahrungen im Formalisieren und Beweisen, Rekursion als Programmieretechnik

Diese Voraussetzungen werden im Modul *Algebra und Diskrete Mathematik* vermittelt.

Verpflichtende Voraussetzungen: Die VO und UE aus dem Modul *Algebra und Diskrete Mathematik*.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Geblockte Einführungsvorlesung (im Gesamtumfang von knapp 1.5h/1.5 ECTS), danach umfangreiche individuell auszuarbeitende Aufgaben zum Argumentieren und Beweisen (im Umfang

von 4.5 ECTS). Ausführliche Präsentation der Beweise (alle Lösungen durch jede Teilnehmerin/jeden Teilnehmer). Exemplarische Ausarbeitung einiger Lösungen, Korrektur durch LVA Leiter/Tutoren zwecks Rückmeldung. Leistungsermittlung auf Grund der Präsentationen und der berechtigten schriftlichen Ausarbeitungen.

Lehrveranstaltungen des Moduls:

6.0/4.0 VU Argumentieren und Beweisen

Bachelorarbeit

Regelarbeitsaufwand: 13.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse: Wissenschaftliche Methodik, internationaler Wissenschaftsbetrieb

Kognitive und praktische Fertigkeiten: Systematische Recherche, Präsentationstechniken, strukturierte und konzise Kommunikation von Inhalten in mündlicher und schriftlicher Form, Fähigkeit zur Anwendung der im Studium erworbenen Kenntnisse und Fertigkeiten im Kontext einer größeren Problemstellung

Soziale Kompetenzen, Innovationskompetenz und Kreativität: Selbstorganisation, Eigenverantwortlichkeit und Eigeninitiative, Teamfähigkeit, Finden kreativer Problemlösungen, Reflexion der eigenen Arbeit im technischen und gesellschaftlichen Kontext

Inhalt: Im Rahmen des Seminars *Wissenschaftliches Arbeiten* lernen die Studierenden wissenschaftliche Methoden und den Wissenschaftsbetrieb kennen. An Hand eines vorgegebenen Themas üben sie Recherche sowie schriftliche und mündliche Präsentation. Darauf aufbauend wenden sie im Projekt *Bachelorarbeit für Informatik und Wirtschaftsinformatik* die im Studium erworbenen Kenntnisse und Fertigkeiten auf ein Thema an, das dem Qualifikationsprofil des Studiums entspricht. Die erzielten Ergebnisse werden neben der Aufgabenstellung, den angewandten Methoden und dem Umfeld in einer schriftlichen Abschlussarbeit dargestellt.

Erwartete Vorkenntnisse: Die Arbeit an der Bachelorarbeit setzt die Kenntnisse, Fertigkeiten und Kompetenzen zumindest der Pflichtmodule dieses Studiums voraus.

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Im Seminar besteht bei den Vorträgen zu Wissenschaftsmethodik und -betrieb sowie bei der Präsentation der Rechercheergebnisse Anwesenheitspflicht, ebenso bei der Präsentation der Bachelorarbeiten. Davon abgesehen können das Seminar- und das Bachelorarbeitsthema in Absprache mit den Lehrenden zeitlich und örtlich weitgehend ungebunden bearbeitet werden. Die Beurteilung orientiert sich an der Qualität und Originalität der mündlichen und schriftlichen Darstellung der Themen sowie der dafür notwendigen Vorarbeiten und berücksichtigt auch das Engagement bei der Diskussion der Arbeiten anderer Studierender.

Lehrveranstaltungen des Moduls:

3.0/2.0 SE Wissenschaftliches Arbeiten

10.0/5.0 PR Bachelorarbeit für Informatik und Wirtschaftsinformatik

Betriebssysteme

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse:

- Kenntnisse über die Rolle und Aufgaben von Betriebssystemen
- Verstehen von Designentscheidungen für das Management von Systemressourcen
- Verständnis der Mechanismen zur Koordination und Synchronisation paralleler Prozesse
- Grundkenntnisse der Netzwerkkommunikation und des Zugriffsschutzes

Kognitive und praktische Fertigkeiten:

- Arbeiten mit Betriebssystemen und Programmierung unter Verwendung von Betriebssystemservices
- Programmierung paralleler Prozesse unter Verwendung gemeinsamer Ressourcen und Nutzung der Kommunikations- und Synchronisationsmechanismen eines Betriebssystems

Soziale Kompetenzen, Innovationskompetenz und Kreativität:

- Erkennen und Einschätzen künftiger Entwicklungen von Betriebssystemtechnologien
- Fähigkeit zur Abstraktion
- Identifikation und Analyse von Kommunikations- und Synchronisationsanforderungen beim Ressourcenmanagement, sowie Lösung entsprechender Synchronisationsaufgaben

Inhalt:

- Grundkonzepte Betriebssysteme
- Prozesse, Threads und Scheduling
- Prozesssynchronisation und Deadlock

- Speicherverwaltung
- Ein/Ausgabe und Disk Management
- Networking
- Security und Protection
- Arbeiten mit Betriebssystemen

Erwartete Vorkenntnisse:

Fachliche und methodische Kenntnisse: Kenntnisse von Zahlendarstellungen in Computern, der grundlegenden Funktionsweise von Computern, endlicher Automaten, Transducer, Grammatiken, Programmiersprachen, sowie Kenntnisse der systematischen Vorgehensweise bei der Programmerstellung.

Kognitive und praktische Fertigkeiten: Interpretieren und Arbeiten mit Zahlendarstellungen und Automaten. Kenntnisse der Programmierung in einer Programmiersprache und der systematischen Programmerstellung und Evaluation.

Soziale Kompetenzen, Innovationskompetenz und Kreativität: Analyse komplexer Zusammenhänge und Wechselwirkungen, Strukturieren und Entwerfen von modularen, interagierenden Systemen.

Diese Voraussetzungen werden in folgenden Modulen vermittelt: *Technische Grundlagen der Informatik, Programmkonstruktion.*

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Das Modul setzt sich aus einem Vorlesungsteil und einer Laborübung zusammen. Die Grundlagen, zentralen Konzepte und theoretischen Inhalte zu Betriebssystemen werden im Vorlesungsteil präsentiert. Ausgewählte Inhalte und Problemstellungen aus dem Bereich der Betriebssystemprogrammierung werden in der Laborübung unter UNIX (Linux) programmiert. Einführungswissen zu den zu lösenden Aufgabenstellungen wird in begleitenden Vortragsblöcken angeboten. Schwerpunkte der Laborübung sind:

- Arbeiten unter Unix/Linux: Shell, Prozesse, Signale, Filesystem
- Programmieren mit der Systemprogrammiersprache C, Debugging
- Systemprogrammierung mit folgenden Mechanismen
 - Parameter und Optionsbehandlung, Filebehandlung
 - Sockets
 - Signale und Signalbehandlung
 - verwandte Prozesse (fork, exec, wait)
 - Kommunikationsmechanismen: Named und Unnamed Pipes, Message Queues

- Synchronisation mit Semaphoren bzw. Sequencer und Eventcounts
- Kommunikation über Shared Memory
- Ressourcenverwaltung

Lehrveranstaltungen des Moduls:

2.0/2.0 VO Betriebssysteme

4.0/2.0 UE Betriebssysteme

Computernumerik

Regelarbeitsaufwand: 4.5 Ects

Bildungsziele: Studenten werden mit den grundlegenden Konzepten algorithmisch-numerischer Lösungsmethoden vertraut gemacht. Die überlegte Auswahl und der effiziente Einsatz kommerzieller oder frei verfügbarer numerischer Software wird vermittelt. Die Studierenden lernen zu erkennen, ob man von einem Programm eine angemessene Lösung erhalten hat und was man tut, wenn dies nicht der Fall ist.

Inhalt: Grundlegende Fehlerbegriffe, Kondition mathematischer Probleme, Datenfehler, Diskretisierungsfehler, Rundungsfehler, Numerische Lösung linearer Gleichungssysteme, numerische Differentiation und Integration, polynomiale Interpolation und Approximation, Design und Verwendung numerischer Algorithmen bzw. numerischer Software.

Die praktische Umsetzung und Vertiefung des Stoffes der Vorlesung erfolgt in der Übung durch realitätsnahe numerische Übungsprojekte. Diese beinhalten sowohl theoretische Aufgabengabenstellungen, etwa was das Design numerisch stabiler Algorithmen betrifft, als auch die praktische Implementierung und das Testen und Bewerten am Computer. Standardsoftware kommt zum Einsatz (z.B. MATLAB).

Erwartete Vorkenntnisse:

Fachliche und methodische Kenntnisse: Mathematische Grundkenntnisse

Kognitive und praktische Fertigkeiten: Programmierung

Diese Voraussetzungen werden in folgenden Modulen vermittelt: *Algebra und Diskrete Mathematik, Analysis, Technische Grundlagen der Informatik, Programmkonstruktion*

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die Vermittlung der theoretischen Grundlagen erfolgt in der Vorlesung, die Erarbeitung der praktische Fertigkeiten erfolgt in Kleingruppen mit regelmäßige Sprechstunden mit dem Betreuer in der Übung.

Die Prüfung ist schriftlich und beinhaltet eher theoretisch gehaltene Fragen zum Vorlesungsstoff, teilweise auch kurz gehaltene Rechenbeispiele; weiters Beurteilung der Übungsleistung.

Lehrveranstaltungen des Moduls:

3.0/2.0 VO Computernumerik

1.5/1.0 UE Computernumerik

Datenbanksysteme

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse:

- Grundlagen, Komponenten, und Funktionsweise von Datenbankmanagementsystemen (DBMS); Datenbankarchitektur und Datenunabhängigkeit
- Komplexe SQL Abfragen, Einbettung in prozedurale Abfragen (JDBC)
- Physische Datenorganisation, Datenbanktuning
- Transaktionen, Fehlerbehandlung, Mehrbenutzersynchronisation
- Verteilte Datenbanken

Kognitive und praktische Fertigkeiten:

- Verwendung von DBMS und Benutzung deklarativer Abfragesprachen
- Programmierung von und Anbindung an Datenbanksysteme

Soziale Kompetenzen, Innovationskompetenz und Kreativität:

- Funktionale Denkweise zum Verständnis deklarativer Abfragesprachen
- Logisches Denken um Abläufe in einem DBMS nachzuvollziehen
- Mathematisch fundierte Vorgehensweise zur Analyse von Methoden in DBMS
- Kenntnisse der eigenen Fähigkeiten und Grenzen, Kritikfähigkeit an der eigenen Arbeit
- Selbstorganisation und Eigenverantwortlichkeit zum eigenständigen Lösen von Laboraufgaben

Inhalt:

- Komponenten und Funktionsweise von Datenbankmanagementsystemen
- Datenbankprogrammierung (komplexe SQL Anfragen, Datenbankanbindung (JDBC), stored procedures)
- Physische Datenorganisation, Anfragebearbeitung und -optimierung
- Transaktionen, Fehlerbehandlung/Recovery, Mehrbenutzersynchronisation
- Verteilte Datenbanken

Erwartete Vorkenntnisse:

Fachliche und methodische Kenntnisse:

- Relationale Modellierung und Algebra
- Grundlegende Kenntnisse von prozeduralen bzw. objektorientierten Programmiersprachen
- Grundlegende Kenntnisse von deklarativen Abfragesprachen
- Relationale Entwurfstheorie (funktionale Abhängigkeiten, Schemadekomposition, Schlüssel, Normalformen)
- Grundlegende Kenntnisse in mathematischer Logik
- Grundlegende Kenntnisse in Graphentheorie

Kognitive und praktische Fertigkeiten: Fähigkeit zum Lesen und Schreiben mathematischer Notationen

Diese Voraussetzungen werden in folgenden Modulen vermittelt: *Modellierung, Programmkonstruktion, Algebra und Diskrete Mathematik*

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung:

- Präsentation der Inhalte in einem Vorlesungsteil
- Laborübungen
- Diskussion mit Tutoren zur Vertiefung des Verständnis

Lehrveranstaltungen des Moduls:

6.0/4.0 VU Datenbanksysteme

Deklaratives Problemlösen

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse: Kenntnisse unterschiedlicher Werkzeuge, Sprachen und logikorientierter Programmiermethoden zum deklarativen Problemlösen sowie deren Grundlagen

Kognitive und praktische Fertigkeiten: Fähigkeit zum Lösen von Problemen mittels verschiedener Werkzeuge basierend auf klassischer Logik und Logikprogrammierung, Kompetenzen zur deklarativen Repräsentation von Problemen

Soziale Kompetenzen, Innovationskompetenz und Kreativität: Verständnis für das Potential deklarativer Methoden, Fähigkeit zur selbständigen Problemlösungskompetenz

Inhalt:

- Grundlagen moderner Entscheidungsprozeduren für die Erfüllbarkeit aussagenlogischer und quantifizierter aussagenlogischer Formeln (SAT und QSAT Solver)
- Normalformtransformationen
- Problemlösen mittels SAT und QSAT Solver
- Systeme und Semantiken der Logikprogrammierung
- Eigenschaften der Antwortmengenprogrammierung
- Praktische Anwendungen der Antwortmengenprogrammierung zur Lösung computationaler Probleme

Erwartete Vorkenntnisse:

Fachliche und methodische Kenntnisse: Grundkenntnisse der Prädikatenlogik (Syntax und Semantik).

Kognitive und praktische Fertigkeiten: analytisches Denken

Soziale Kompetenzen, Innovationskompetenz und Kreativität: Fähigkeit zur selbständigen Einarbeitung in Werkzeuge anhand schriftlicher Dokumentationen

Diese Voraussetzungen werden im Modul *Modellierung* vermittelt.

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die Lehrveranstaltungen des Moduls bestehen aus einer Vorlesung und einer Laborübung. Die Vorlesung dient zur Vermittlung der theoretischen Grundlagen des besprochenen Fachgebietes während in der Laborübung die Teilnehmer in selbständiger Weise Lösungen zu konkreten Aufgabenstellungen erarbeiten. Die Beurteilung erfolgt auf Basis von Prüfungen (schriftlich und/oder mündlich) sowie der abgegebenen Lösungen der Aufgabenstellungen.

Lehrveranstaltungen des Moduls:

3.0/2.0 VO Deklaratives Problemlösen

3.0/2.0 UE Deklaratives Problemlösen

Einführung in Visual Computing

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse: Das Modul vermittelt grundlegende Kenntnisse im Bereich Visual Computing und ein kritisches Verständnis ihrer Theorien und Grundsätze:

- Computergraphik,
- Computer Vision,
- Digitale Bildverarbeitung,
- Visualisierung,
- Augmented/Mixed/Virtual Reality.

Kognitive und praktische Fertigkeiten: Durch die praktische Auseinandersetzung mit aktuellen Technologien, Methoden und Werkzeugen (wie modernen Programmiersprachen und Entwicklungsumgebungen) werden folgende kognitiven Fertigkeiten vermittelt:

- Einsatz formaler Grundlagen und Methoden zur Modellbildung, Lösungsfindung und Evaluation,
- Methodisch fundierte Herangehensweise an Probleme, insbesondere im Umgang mit offenen/unspezifizierten Problemsituationen,
- Entwurfs- und Implementierungsstrategien.

Soziale Kompetenzen, Innovationskompetenz und Kreativität: Der Schwerpunkt liegt hier auf der besonderen Förderung hoher Kreativitäts- und Innovationspotentiale.

- Eigeninitiative und Neugierde,
- Selbstorganisation, Eigenverantwortlichkeit,
- Problemformulierungs- und Problemlösungskompetenz,
- Kenntnisse der eigenen Fähigkeiten und Grenzen, Kritikfähigkeit.

Inhalt:

- Digitale Bilder: Auflösung, Abtastung, Quantisierung, Farbrepräsentation
- Bildoperationen: Punktoperationen, lokale und globale Operationen
- Segmentierung

- Bewegungserkennung
- Repräsentation: konturbasiert, regionenbasiert (Momente, Graphen)
- Kodierung: Entropie-Kodierung, Source-Kodierung
- Komprimierung: Prediktive Kodierung, Vektorquantisierung, JPEG, MPEG
- Hardware: Ein- und Ausgabegeräte, Bildgebende Verfahren, Sensoren
- Radiometrische und Geometrische Transformationen
- Graphikprimitive und deren Attribute
- 2D- und 3D-Viewing, Graphikarchitektur (Rendering Pipeline, etc)
- Sichtbarkeitsverfahren
- 3D Objektrepräsentationen
- Kurven und Flächen
- Licht und Schattierung
- Ray-Tracing und Globale Beleuchtung
- Texturen und andere Mappings
- Farben und Farbmodelle
- Computational Photography
- Non-photorealistic Rendering

Erwartete Vorkenntnisse:

- Mathematik auf Maturaniveau (Vektorrechnung, Winkelfunktionen, Differenzieren und Integrieren, lineare Algebra, einfache Geometrie)
- Elementare Programmierkenntnisse

Diese Voraussetzungen werden in folgenden Modulen vermittelt: *Algebra und Diskrete Mathematik, Programmkonstruktion*

Verpflichtende Voraussetzungen: Keine.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Vorlesung mit Übung: Angesichts der großen Anzahl von HörerInnen wird das eine Frontalvorlesung mit Unterstützung durch Medien (hauptsächlich Datenprojektor) sein, in die Übungsbeispiele eingebaut sind. Es soll ein kompaktes Skriptum geben. Die Leistungsbeurteilung wird durch eine Prüfung erfolgen, wobei u.U. die erfolgreiche Teilnahme an einzelnen Übungsteilen vorausgesetzt werden könnte.

Lehrveranstaltungen des Moduls:

6.0/5.0 VU Einführung in Visual Computing

Entwicklung von Web-Anwendungen

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse:

- Das Modul vermittelt ein breites und integriertes Wissen und Verstehen der wissenschaftlichen Grundlagen der Entwicklung von Web-Anwendungen, welches wesentlich über das auf der Ebene der Universitätszugangsberechtigung vorhandene Wissen hinausgeht.
- Die Studierenden sind befähigt Problemdomänen zu beherrschen, die sich durch wenig strukturierte Information auszeichnen.
- Die Studierenden verfügen über ein kritisches Verständnis der wichtigsten Theorien, Prinzipien und Konzepte zur Entwicklung von Web-Anwendungen.
- Das Wissen und Verständnis der Studierenden entspricht dem Stand der Fachliteratur in diesem Bereich.

Kognitive und praktische Fertigkeiten:

- Die Studierenden können ihr Wissen und Verstehen praktisch zur Lösung von Aufgaben im Bereich der Web-Anwendungsentwicklung umsetzen.
- Die Studierenden können für die jeweilig vorliegende Aufgabenstellung relevante Informationen sammeln, strukturieren, bewerten und interpretieren.

Soziale Kompetenzen, Innovationskompetenz und Kreativität:

- Die Studierenden sind in der Lage ihr Wissen selbständig zu vertiefen.
- Die Studierenden können entwicklungsbezogene Positionen und Problemlösungen formulieren, sich mit InformatikerInnen und DomänenexpertInnen darüber austauschen und Verantwortung in einem Team übernehmen.
- Die Studierenden lernen ihre eigenen Fähigkeiten und Grenzen einzuschätzen und erwerben die Kritikfähigkeit an der eigenen Arbeit.
- Die Studierenden erlernen Selbstorganisation und Eigenverantwortlichkeit zum eigenständigen Lösen von Aufgaben.

Inhalt: Grundlagen

- Begriffsdefinitionen
- Grundlagen semistrukturierter Daten

- Architekturelle Grundlagen des World Wide Web
- Grundlagen von Web-Anwendungen

Sprachen und Technologien

- Grundlagen von Markup-Sprachen
- Schemasprachen
- Abfragesprachen
- Web-Modellierungssprachen
- Technologien für dynamische Web-Anwendungen
- Web Services

Umsetzung und praktische Realisierung

- Barrierefreie Web-Anwendungen
- Entwicklungsprozess
- Entwurfsmuster
- Entwicklungs-APIs
- Stylesheets

Erwartete Vorkenntnisse:

Es werden die Fertigkeiten und Kenntnisse folgender Module benötigt: *Modellierung, Programmkonstruktion*

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die Lehrveranstaltung Semistrukturierte Daten besteht aus Vorlesungs- und Übungsteil. Es gibt 3 Übungseinheiten, die die Konzepte der Vorlesung vertiefen. Davor muss ein obligatorischer Einstiegstest positiv absolviert werden. Die Übungsbeispiele werden, bevor sie starten, in der Vorlesung besprochen. Zur Betreuung der Übung gibt es Fragestunden bei den Tutorinnen und Tutoren. Die Übung besteht hauptsächlich aus Programmieraufgaben, die die Studierenden selbständig zu lösen haben. Die Übungsleistung fließt zu 25 Prozent in die Gesamtnote ein. Um für alle Studierenden gleiche Voraussetzungen bei der Abgabe zu schaffen, gibt es einen einheitlichen Abgabetermin für alle Studierenden, gefolgt von den Abgabegesprächen. Bei den Abgaben werden sowohl die Beispiele auf Korrektheit geprüft als auch das Verständnis der Studierenden bewertet. Die Prüfung besteht aus theoretischen und praktischen Beispielen. Es gibt vier Prüfungstermine (einen am Semesterende, drei weitere im Folgesemester), die die verbleibenden 75 Prozent der Note ausmachen. Die Lehrveranstaltung Web Engineering umfasst

Vorlesungs- und Übungsteil. Der Inhalt der Vorlesung wird in 4 praktischen Übungsaufgaben vertieft, welche in 3er Gruppen gelöst werden. Vor der Gruppenbildung muss ein obligatorischer Einstiegstest positiv absolviert werden. Pro Übungsaufgabe müssen mindestens 30 Beide Lehrveranstaltungen basieren ihre Übungsangaben auf derselben Anwendungsdomäne. Dies soll einerseits für die Studierenden ein durchgängiges Übungsszenario schaffen und andererseits das Verständnis für die Methodenvielfalt im Bereich der Web-Anwendungsentwicklung schärfen.

Lehrveranstaltungen des Moduls:

3.0/2.0 VU Semistrukturierte Daten

3.0/2.0 VU Web Engineering

Fachübergreifende Qualifikationen

Regelarbeitsaufwand: 9.0 Ects

Bildungsziele: Durch dieses Modul sollen Studierende Qualifikationen erwerben, die über die für das Studium typischen fachlichen Kenntnisse und Fertigkeiten hinausgehen und im Berufsalltag eine wesentliche Rolle spielen, wie zum Beispiel: Verhandlungsführung, Präsentations- und Kommunikationstechnik, systematische Recherche und Planung, Konfliktmanagement, Teamfähigkeit und Führung, Organisation und Management, Betriebsgründung und Finanzierung, Verständnis rechtlicher Rahmenbedingungen, Verbesserung von Fremdsprachenkenntnissen.

Lehrveranstaltungen des Moduls: Die Lehrveranstaltungen dieses Moduls sind im Umfang von mindestens 9.0 Ects aus dem von der Technischen Universität Wien verlautbarten Katalog von Lehrveranstaltung zum Erwerb von fachübergreifenden Qualifikationen sowie aus den folgenden Lehrveranstaltungen zu wählen.

3.0/2.0 SE Coaching als Führungsinstrument 1

3.0/2.0 SE Coaching als Führungsinstrument 2

3.0/2.0 SE Didaktik in der Informatik

1.5/1.0 VO EDV-Vertragsrecht

3.0/2.0 VO Einführung in die Wissenschaftstheorie I

3.0/2.0 VO Einführung in Technik und Gesellschaft

3.0/2.0 SE Folgenabschätzung von Informationstechnologien

3.0/2.0 VU Forschungsmethoden

3.0/2.0 VO Frauen in Naturwissenschaft und Technik

3.0/2.0 SE Gruppendynamik

3.0/2.0 VU Italienisch für Ingenieure I

3.0/2.0 VU Kommunikation und Moderation

3.0/2.0 SE Kommunikation und Rhetorik

1.5/1.0 SE Kommunikationstechnik

3.0/2.0 VU Kooperatives Arbeiten

1.5/1.0 VO Präsentation, Moderation und Mediation

3.0/2.0 UE Präsentation, Moderation und Mediation

3.0/2.0 VU Präsentations- und Verhandlungstechnik
3.0/2.0 SE Rechtsinformationsrecherche im Internet
3.0/2.0 VU Rhetorik, Körpersprache, Argumentationstraining
3.0/2.0 VU Softskills für TechnikerInnen
3.0/2.0 VU Technical English Communication
3.0/2.0 VU Technical English Presentation
3.0/2.0 VU Techniksoziologie und Technikpsychologie
3.0/2.0 VU Technisches Französisch, Hohes Niveau I
3.0/2.0 VU Technisches Russisch I
3.0/2.0 VU Technisches Russisch II
3.0/2.0 VU Technisches Spanisch I
3.0/2.0 VU Technisches Spanisch II
3.0/2.0 VO Theorie und Praxis der Gruppenarbeit
3.0/2.0 VO Zwischen Karriere und Barriere

Freie Wahl

Regelarbeitsaufwand: max. 9.0 Ects

Bildungsziele: Die Lehrveranstaltungen dieses Moduls dienen der Vertiefung des Faches sowie der Aneignung außerfachlicher Kenntnisse, Fähigkeiten und Kompetenzen.

Lehrveranstaltungen des Moduls: Die Lehrveranstaltungen dieses Moduls können frei aus dem Angebot an wissenschaftlichen/künstlerischen Lehrveranstaltungen aller anerkannten in- und ausländischen Universitäten gewählt werden, sofern sie der Vertiefung des Faches oder der Aneignung außerfachlicher Kenntnisse, Fähigkeiten und Kompetenzen dienen. Der Umfang der frei wählbaren Lehrveranstaltungen ergänzt den Umfang der übrigen im Studium absolvierten Lehrveranstaltungen auf 180 Ects (oder mehr), wobei ihr Anteil daran 9.0 Ects nicht übersteigen darf.

Grundlagen der Human Computer Interaction

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele: The aim of this module is to give a foundation for students to understand how computing and technology and the design decisions that they will make as Informatics people impact 1. the world and society at large and 2. the interactive experiences needing to use technology. Students should also have a basic understanding of the fundamental concepts, skills and processes underlying the design of Technik für Menschen (as per the TU mission statement).

Kognitive und praktische Fertigkeiten: By the end of this module successful students will be able to:

- Critically reflect on the social, cognitive and cultural impact of technology in the world

- Understand the principles of design and evaluation methods
- Cope with open and unspecified problem situations in a more systematic and constructive way
- Understand the value of interdisciplinary and system-oriented approaches
- Apply basic HCI techniques to plan and implement common design strategies
- Understand human behaviour and perception and how it relates to interaction
- Understand human perspectives of systems design
- Understand the principle of usable, useful and engaging systems
- Critically reflect on what makes good and bad design
- Understand the principles of a people-centred design process
- Understand the diverse skills of experts in a software development process.
- Understand the necessity for a design-based approach to the implementation of interactive systems, and relate their skills and expertise to the necessities of such an approach.

Soziale Kompetenzen, Innovationskompetenz und Kreativität: Students should learn to:

- Critically read and discuss scientific literature and understand the public discourse related to computers and society;
- Work in teams on reflection and design problems;
- Be able to discuss technologies and needs with potential users;
- Come up with innovative ideas for interactive technologies;
- Approach open and ambiguous problem situations in a proactive and self-organized way.
- Reflect on their own work in a social, cognitive and cultural context.

Inhalt: Broad content areas include:

- History of ideas, technologies, markets and organizations in informatics
- Dissection of the interactions between technology and society
- Examination of exemplary areas of tension between ICT and society, such as copyright, patents, vulnerability, privacy and security from multiple perspectives

- Discussion of current and past developments and trends in the aforementioned areas
- Theories of Human perception, cognition and practice
- Theoretical foundation of user experience
- Design principles and interface design guidelines
- Conducting usability studies and expert evaluations of interactive systems
- Principles of a user centred interaction design process
- HCI related to different types of interactive software

Verpflichtende Voraussetzungen: Keine.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Courses in the module are organized to require student to work in a self-organized, self-dependent and reflective way. Based on a suitable presentation of content (eg. lectures) students choose from a catalog of predefined suitable activities and exercises in order to hand in relevant, in-depth work that shows their learning progress. These activities include, but are not limited to, reading and reflecting upon scientific literature, doing well-defined practical exercises, discussing related issues with outside experts, peer-reviewing of other's work, etc.

Lehrveranstaltungen des Moduls:

3.0/2.0 VU Gesellschaftliche Spannungsfelder der Informatik

3.0/2.0 VU Basics of Human Computer Interaction

Grundlagen intelligenter Systeme

Regelarbeitsaufwand: 8.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse: Das Modul vermittelt:

- Grundkenntnisse in den theoretischen Grundlagen intelligenter Systeme
- Fundamentale Konzepte, die zum Verständnis der Arbeitsweise als auch der Erstellung intelligenter Systeme von Bedeutung sind

Kognitive und praktische Fertigkeiten:

- Fähigkeit zur formalen Analyse der eingesetzten Techniken und Methoden
- Fähigkeit, Methoden und Techniken für eine vorgegebene Aufgabenstellung auszuwählen

- Fähigkeit zur Analyse einfacher Aufgabenstellungen sowie die Umsetzung von Lösungen in eine geeignete Form der Wissensrepräsentation mit dazugehörigem Verarbeitungsmechanismus
- Kritische Bewertung und Reflexion von Lösungen
- Präsentation von Lösungen

Soziale Kompetenzen, Innovationskompetenz und Kreativität:

- Selbstorganisation, Eigenverantwortlichkeit
- Eigeninitiative und Neugierde
- Umgang mit formalen Beschreibungen und darauf aufsetzenden Verarbeitungsmechanismen

Inhalt: Einführung in die Künstliche Intelligenz:

- Einführung und Geschichte
- Suchverfahren
- Constraint Satisfaction Probleme (CSP)
- Planen
- Lernen
- Neuronale Netze
- Entscheidungstheoretische Konzepte
- Philosophische Aspekte der KI

Einführung in wissensbasierte Systeme:

- Einführung und geschichtlicher Hintergrund
- Die Architektur eines wissensbasierten Systems
- Formalismen zur Wissensrepräsentation
- Deduktionskonzepte korrespondierend zu den Formalismen
- Implementierungen nichtmonotonen Verhaltens
- Modellbasierte Diagnose
- Konzepte der Unsicherheit

- Entwicklung von wissensbasierten Systemen, moderne Anwendungen etwa in der Geodäsie

Erwartete Vorkenntnisse:

Fachliche und methodische Kenntnisse:

- Elementare Kenntnisse der Aussagen- und Prädikatenlogik
- Algorithmen
- Fähigkeit, einfache Beweise zu führen

Diese Voraussetzungen werden in folgenden Modulen vermittelt: *Algebra und Diskrete Mathematik, Algorithmen und Datenstrukturen, Modellierung*

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung:

- Präsentation der Lehrinhalte in einem Vorlesungsteil (Frontalvortrag)
- Selbständiges Ausarbeiten von Aufgabenstellungen durch Studierende
- Präsentation der Lösungen (inkl. der benötigten Theorie)

Leistungsbeurteilung:

- Schriftliche Prüfung und
- Bewertung der Ausarbeitungen (inkl. der Präsentation)

Lehrveranstaltungen des Moduls:

3.0/2.0 VU Einführung in die Künstliche Intelligenz

5.0/3.0 VU Einführung in wissensbasierte Systeme

Kontexte der Systementwicklung

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele: The aim of this module is to create an understanding of the importance of the contexts of society, ethics and design as well as human work and recreation for software development. It will enable students to understand the aspects of software development beyond the issues addressed from technical and processual perspectives. Students should have an in-depth understanding of the concepts, skills and processes to face the inconsistent, diverse and ambiguous requirements of human society and individuals in software development.

Fachliche und methodische Kenntnisse: By the end of this module successful students will be able to:

- Incorporate the critical reflection on the social, cognitive and cultural impact of technology in the world into the design of interactive systems
- Exercise basic design thinking and design practice in complex and ambiguous problem settings
- Apply basic qualitative or participative approaches for problem setting and problem solving
- Approach open and unspecified problem situations in a constructive way
- Apply advanced HCI techniques to plan and implement common design strategies
- Develop and apply design and implementation strategies appropriate for the diverse and complex needs of human work and recreation
- Understand the complex contexts of real world problem situations
- Implement usable, useful and engaging systems
- Plan and implement appropriate human-centered approaches in the implementation of interactive systems
- Form a broad and sustainable understanding for a design process that takes into account the constraints set by society, organizations, individuals as well as the designer.

Soziale Kompetenzen, Innovationskompetenz und Kreativität: Students should learn to

- Contextualize knowledge,
- Acquire more in depth knowledge of the general characteristics of work, organization and leisure activities for which they will design interactive systems, as well as methods and theories for the acquisition of such knowledge
- Understand the value of analyzing use situations of technology for design.
- Work in teams on reflection and design problems
- Understand the interplay between problem setting and problem solving
- Be able to discuss technologies and needs with potential users, and transform the results of these discussions into designs
- Come up with innovative ideas for all kinds of information and communication technologies;
- Deal with open and ambiguous problem situations in a constructive and holistic manner.

- Constructively embed their work into a social, cognitive and cultural context.
- Take responsibility for the design of interactive systems in complex projects

Inhalt: Broad content areas include:

- Social and ethical issues of information and communication technologies
- User experience and interaction design
- Participatory design
- Methods of User Research and Evaluation

Erwartete Vorkenntnisse:

Dieses Modul baut auf den Kenntnissen, Fertigkeiten und Kompetenzen folgender Module auf: *Grundlagen der Human Computer Interaction, Modellierung, Programmkonstruktion, Software Engineering und Projektmanagement*

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: lectures with accompanied exercises/tutorials

Lehrveranstaltungen des Moduls:

- 3.0/2.0 VU Gesellschaftswissenschaftliche Grundlagen der Informatik
- 3.0/2.0 VU Interface and Interaction Design

Logikprogrammierung und Constraints

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse:

- Programmiersprache Prolog
- Constraintsprachen CLP(FD), CHR.
- Spezifikationsorientierte Programmierung
- Deklarative Diagnose

Kognitive und praktische Fertigkeiten: Durch das praktische Arbeiten mit einer logikorientierten Programmiersprache werden die folgenden Fertigkeiten vermittelt:

- Deklaratives Modellieren, relationale Sichtweise
- Praktische Programmierkenntnisse in einer logikorientierten Programmiersprache

- Anwendung deklarativer Lesarten zur Fehlersuche
- Verbindung und Abwägung von deklarativen und prozeduralen Sichtweisen

Soziale Kompetenzen, Innovationskompetenz und Kreativität:

- Der Übungsbetrieb fördert das selbständige Arbeiten in Eigenverantwortlichkeit
- Mittels des logikorientierten Programmierparadigmas wird eine neue Sichtweise des Programmierens ermöglicht
- Zusammenarbeit, insbesondere bei Anwendung der Lesarten

Inhalt:

- Deklarative Programmierparadigmen
- Deklarative Lesarten
- Deklarative Diagnose
- Prozedurale Lesarten
- Termination
- Grammatiken
- Constraints
- Programmieren höherer Ordnung
- Lambda-Ausdrücke
- Pure I/O

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Der Vorlesungsteil wird parallel zum Übungsteil abgehalten, sodass auch die konkrete Vorgangsweise beim Programmieren und der deklarativen Fehlersuche behandelt werden kann. Der Übungsteil besteht aus vielen kleinen Beispielen. Die Leistungsbeurteilung besteht aus einer prüfungsimmanenten Beurteilung der Programmierfähigkeit und einem mündlichen Abgabegespräch. Inhalt des Abgabegesprächs sind die Lesarten von Logikprogrammen anhand konkreter Beispiele sowie deren Anwendung zur Fehlersuche.

Lehrveranstaltungen des Moduls:

6.0/4.0 VU Logikprogrammierung und Constraints

Microcontroller und Betriebssysteme

Regelarbeitsaufwand: 10.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse:

- Grundlagen Microcontroller-Architekturen und -Komponenten
- (Mikro-)Elektronik-Kennnisse für System Design, IO-Interfaces
- Einblick in die praktische Umsetzung von Betriebssystemkonzepten
- Kenntnisse einer systematischen Vorgehensweise zur Implementierung von Betriebssystemkomponenten

Kognitive und praktische Fertigkeiten:

- Fertigkeiten in der Programmierung von Microcontrollern
- Verständnis von Datenblättern
- Wissenschaftlich fundierte Problemlösung: Analyse von Buffergrößen, Reaktionszeiten von Systemen, etc.
- Fertigkeiten in Design und der Implementierung von Betriebssystemkomponenten
- Fehlersuche und -beseitigung im Kontext von Systemsoftware
- Verständnis für Aspekte und Probleme unterhalb der Abstraktion von Hochsprachen

Soziale Kompetenzen, Innovationskompetenz und Kreativität:

- Interdisziplinäre und flexible/anpassungsfähige Denkweise: Von 'normalen' Systemen zu Embedded Systems.
- Selbstständige und umfassende Lösung von Aufgabenstellungen.

Inhalt:

- Microcontroller-Architekturen
- Peripheriemodule von Microcontrollern
- Serielle und parallele digitale Kommunikation
- Analog-Digital- und Digital-Analog-Interfaces
- Entwicklungs-Toolchain

- Microcontroller-Programmierung in Assembler und einer Hochsprache (C)
- Microcontroller-Programmierung unter einem Embedded Systems OS.
- Systematisches Debugging in Embedded Systems
- Interne Struktur eines Betriebssystems
- Implementierung ausgewählter Betriebssystemkomponenten aus dem Bereichen
 - Speicherverwaltung (virtueller Speicher)
 - Prozessverwaltung (Scheduling, Synchronisation)
 - Dateisysteme (Blockverwaltung, Dateiabstraktion)
 - Systemaufruf-Schnittstelle aus der Sicht des Betriebssystementwicklers

Erwartete Vorkenntnisse:

Fachliche und methodische Kenntnisse:

- Physikalische Grundlagen elektrischer Komponenten und elektrischer Netzwerke
- Analyse elektrischer Netzwerke
- Wissen über und Verständnis von elementaren elektrischen Schaltungen
- Fundierte Kenntnisse von Zahlendarstellungen in Computern
- Grundlagen der Boole'schen Algebra und Schaltlogik
- Aufbau digitaler Schaltungen, Implementierung von Registern, Speicher und Logik
- Fundierte Kenntnisse in Rechnerarchitekturen
- Fundamentale Algorithmen und Datenstrukturen
- Methoden zur Bewertung und Analyse von Algorithmen

Kognitive und praktische Fertigkeiten:

- Fähigkeit zur Abstraktion elektrischer Vorgänge
- Fertigkeiten in der Verwendung der Rechenverfahren der Elektrotechnik
- Fertigkeiten bei der praktischen Realisierung einfacher elektrischer Schaltungen und deren messtechnischer Untersuchung
- Fähigkeit zum praktischen Entwurf und zur Modellierung von sequenziellen und parallelen Systemen
- Abstrakte und effizienzorientierte Denkweise für die Entwicklung von Programmen

- Fähigkeit zum Einsatz formaler und informeller Methoden zur Spezifikation, Modellierung und Analyse von Algorithmen
- Fundierte Fertigkeiten in der Programmierung in C
- Kenntnisse über die Programmentwicklung unter Unix und Linux

Soziale Kompetenzen, Innovationskompetenz und Kreativität:

- Selbstorganisation und Eigenverantwortlichkeit
- Interdisziplinäres Denken

Diese Voraussetzungen werden in folgenden Modulen vermittelt: *Algebra und Diskrete Mathematik, Analysis, Technische Grundlagen der Informatik, Algorithmen und Datenstrukturen*

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Der erste Teil des Moduls (Microcontroller) setzt sich aus einer Laborübung mit begleitender Vorlesung zusammen, wobei in der einstündigen Vorlesung die theoretischen Konzepte erarbeitet und in der Laborübung die Praktischen Übungsbeispiele gelöst werden. In beiden gibt es mehrere Tests. In der Laborübung gibt es die Möglichkeit, vorgegebene Projekte auszuarbeiten. Der zweite Teil des Moduls (Programmierung von Betriebssystem-Komponenten) ist als Laborübung organisiert, in der in Kleingruppen selbstständige Softwareentwicklung betrieben wird und eine schriftliche Dokumentation relevanter Designentscheidungen und Erkenntnisse zu verfassen ist.

Lehrveranstaltungen des Moduls:

7.0/7.0 VU Microcontroller

3.0/2.0 UE Programmierung von Betriebssystemen

Modelle und Modellierung von statistischen Daten

Regelarbeitsaufwand: 7.5 Ects

Bildungsziele: Mathematische Beschreibungen unscharfer Mengen und Größen, Elemente der Fuzzy Logik, Selbständiges Arbeiten an größeren statistischen Auswertungen

Fachliche und methodische Kenntnisse:

- Verallgemeinerte Mengenoperationen
- Unscharfe Zahlen und unscharfe Vektoren
- t-Normen und t-Conormen
- Kombination unscharfer Zahlen

- Fuzzy Relationen
- Unscharfe Beziehungen zwischen Variablen
- Approximatives Schließen
- Exploration und Analyse statistischer Daten
- Computergestützte Verfahren der Statistik, Schätzungen, statistische Tests, Varianzanalyse, Regression
- Kritische Sicht im Umgang und Anwendung mit statistischer Software

Kognitive und praktische Fertigkeiten:

- Anwendung von Fuzzy Modellen auf verschiedene Problemstellungen
- Anwendung von statistischen Methodiken mit Hilfe des Computers auf konkrete Problemstellung

Soziale Kompetenzen, Innovationskompetenz und Kreativität:

- Umsetzung stochastischer Modelle in konkrete Anwendungen
- Fuzzy Entscheidungsmodelle
- Fuzzy Regler
- Fuzzy Stochastik
- Fuzzy Klassifikation
- Umsetzung von konkreten Aufgaben in statistische und computerverarbeitfähige Problemstellungen
- Entsprechende Lösungsvorgänge

Inhalt: Dieses Modul vermittelt im Einzelnen folgende Themen:

Teilmengen und unscharfe Teilmengen, verallgemeinerte Mengenoperationen, t-Normen und t-Cornormen, Erweiterungsprinzip, rechnen mit unscharfen Zahlen, unscharfe reelle Funktionen, Kombination unscharfer Zahlen zu unscharfen Vektoren, Fuzzy Relationen, unscharfe Beziehungen zwischen Variablen, approximatives Schließen, Fuzzy Maße und Fuzzy Wahrscheinlichkeitsverteilungen, Fuzzy Regler, Fuzzy Stochastik, Fuzzy Zuverlässigkeit, Fuzzy Optimierung, Fuzzy Klassifikation, Fuzzy Entscheidungsmodelle; Exploration und Analyse statistischer Daten, computergestützte Verfahren der Statistik, Vertiefung in und kritischer Vergleich von kommerzieller Statistiksoftware (SPSS, SAS, SPLUS, S, R). Einsatz von spezieller Software, insbesondere Neuentwicklungen, Schwerpunkt liegt aber auf R. Arbeiten mit größeren Fallbeispielen. Der Methodenkatalog

umfasst: Beschreibende Statistik, Vergleich von Gruppen von Daten, Varianzanalyse, Regressionsanalyse, Geostatistik, Ausblick in Multivariate Methoden, Simulation.

Erwartete Vorkenntnisse: Grundlegende Kenntnisse der Mathematik und Statistik

Diese Voraussetzungen werden in folgenden Modulen vermittelt: *Algebra und Diskrete Mathematik, Analysis, Statistik und Wahrscheinlichkeitstheorie*

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die Lehrveranstaltung Fuzzy Modelle ist eine Vorlesungsübung (VU) in der integriert Konzepte vorgestellt und Anwendungen besprochen werden. Die Beurteilung erfolgt einerseits durch laufende Mitarbeit und zusätzlich mit einer mündlichen Abschlussprüfung.

Die Lehrveranstaltung Computerstatistik ist ein Mix aus einem theoretischen Vorlesungsteil und einem Übungsteil, wobei die Teile in natürlicher Weise ineinander übergehen. Die theoretischen Darstellungen werden möglichst zeitnahe mit praktischen Übungen am Computer erläutert und geübt.

Lehrveranstaltungen des Moduls:

3.0/2.0 VU Fuzzy Modelle

4.5/3.0 VU Computerstatistik

Modellierung

Regelarbeitsaufwand: 9.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse:

- Das Modul vermittelt ein breites und integriertes Wissen und Verstehen der wissenschaftlichen Grundlagen der Modellierung.
- Die Studierenden erwerben ein kritisches Verständnis der wichtigsten Theorien, Prinzipien und Konzepte der Modellierung auf dem Stand der Fachliteratur im diesem Bereich.

Kognitive und praktische Fertigkeiten:

- Die Studierenden können ihr Wissen und Verstehen praktisch in Modellierungsaufgaben anwenden und Problemlösungen und Argumente für Modellierungsaufgaben erarbeiten und weiterentwickeln.
- Die Studierenden können die für die Modellierung relevanten Informationen sammeln, bewerten und interpretieren.

Soziale Kompetenzen, Innovationskompetenz und Kreativität:

- Die Studierenden sind in der Lage ihr Wissen selbständig zu vertiefen.

- Die Studierenden können modellierungsbezogene Positionen und Problemlösungen formulieren, sich mit InformatikerInnen und DomänenexpertInnen darüber austauschen und Verantwortung in einem Team übernehmen.
- Die Studierenden lernen ihre eigenen Fähigkeiten und Grenzen einzuschätzen und erwerben die Kritikfähigkeit an der eigenen Arbeit.
- Die Studierenden erlernen Selbstorganisation und Eigenverantwortlichkeit zum eigenständigen Lösen von Aufgaben.

Inhalt:

- Formale Modellierung: Automaten, reguläre Ausdrücke, Grammatiken, Petri-Netze, Aussagen- und Prädikatenlogik als Spezifikations Sprachen, Syntax und Semantik, Modellbegriff
- Datenmodellierung: Grundlagen der Modellierung, Datenbankentwurf, relationales Modell, Datenintegrität
- Objektorientierte Modellierung: Grundlagen objektorientierter Modellierung, Klassen-, Sequenz-, Zustands-, Aktivitäts- und Anwendungsfalldiagramme

Erwartete Vorkenntnisse: Keine.

Verpflichtende Voraussetzungen: Keine.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die Inhalte werden in Vorlesungsblöcken vorgestellt und in begleitenden Übungen von den Studierenden erarbeitet. Die Übungsaufgaben können zeitlich und örtlich weitgehend ungebunden einzeln oder in Gruppen gelöst werden. Die Lösungen werden bei regelmäßigen Treffen mit Lehrenden und TutorInnen besprochen und korrigiert. Die Beurteilung erfolgt auf Basis schriftlicher Tests und der kontinuierlich in den Übungen erbrachten Leistungen. Der Übungsbetrieb und die Tests können computerunterstützt durchgeführt werden.

Lehrveranstaltungen des Moduls: Es sind die folgenden Lehrveranstaltungen zu absolvieren:

- 3.0/2.0 VU Formale Modellierung
- 3.0/2.0 VU Datenmodellierung
- 3.0/2.0 VU Objektorientierte Modellierung

Multivariate und computerintensive statistische Methoden

Regelarbeitsaufwand: 9.0 Ects

Bildungsziele: Vermittlung von datenorientierten, computerintensiven Methoden zur Verarbeitung komplexer Daten

Fachliche und methodische Kenntnisse:

- Multivariate Methoden
- Grundlagen der statistischen Simulation

Kognitive und praktische Fertigkeiten:

- Anwendung multivariater Methoden auf konkrete Problemstellungen
- Anwendung von computerintensiven Methoden und Software auf komplexe Problemstellungen

Soziale Kompetenzen, Innovationskompetenz und Kreativität:

- Datenorientierte Lösung von statistischen Problemstellungen
- Lösungen von Problemen mit open-source Software

Inhalt: Clusteranalyse, Hauptkomponenten- und Faktorenanalyse, Diskriminanzanalyse, Zufallszahlengeneratoren und Reproduzierbarkeit, MCMC (Markov Chain Monte Carlo) Methoden, Resamplingverfahren (Bootstrap, Jackknife, Kreuzvalidierung), Testen mittels statistischer Simulation, Anwendungen von Resamplingverfahren in Zeitreihen, Datenimputation und Regression.

Erwartete Vorkenntnisse: Grundlegende Kenntnisse der computerorientierten Statistik

Diese Voraussetzungen werden im Modul *Statistik und Wahrscheinlichkeitstheorie* vermittelt.

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: In der Lehrveranstaltung Multivariate Statistik werden die gängigen multivariaten Methoden formal vermittelt, und mit der Statistiksoftware R an konkreten Daten angewandt sowie Ergebnisse diskutiert.

In der Lehrveranstaltung Statistische Simulation und computerintensive Methoden (VU) werden simulationsbasierte Lösungsstrategien für komplexe Problemstellungen gelehrt. Die Verfahren der statistischen Simulation und computerintensiver Methoden werden theoretisch als auch praktisch mittels moderner freier open-source Statistiksoftware (R) vermittelt.

Lehrveranstaltungen des Moduls:

4.5/3.0 VO Multivariate Statistik

1.5/1.0 UE Multivariate Statistik

3.0/2.0 VU Statistische Simulation und computerintensive Methoden

Parallel Computing

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele: Grundlegendes Verständniss für Beweggründe und Ziele des parallelen Rechnens, grundlegende Kenntnisse von parallelen Architekturen, Programmiermodellen, Sprachen und Schnittstellen, elementare Fähigkeiten des Programmierens in einer oder mehreren dieser Schnittstellen/Sprachen.

Inhalt: Motivation, Beweggründe, (Geschichte) des parallelen Rechnens. Parallele Rechnerarchitekturen, Programmiermodelle, Leistungsanalyse und -Beurteilung, Einführung in MPI (Message-Passing Interface) Einführung in OpenMP Sprachen und Schnittstellen für Mehrkernrechner

Erwartete Vorkenntnisse: Grundlagen von Programmiersprachen, Rechnerarchitekturen und Betriebssystemen. Programmierkenntnisse (bevorzugt C, C++, Fortran, oder Java)

Fachliche und methodische Kenntnisse: Grundlagen von Programmiersprachen, Rechnerarchitekturen und Betriebssystemen

Kognitive und praktische Fertigkeiten: Programmierkenntnisse

Dieses Modul baut auf den Kenntnissen und Fertigkeiten folgender Module auf: *Betriebssysteme, Programmierparadigmen, Programmkonstruktion, Technische Grundlagen der Informatik*

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Vortrag Übungen an der Tafel, selbständige Bearbeitung von Themen, Programmierübungen und -projekte.

Beurteilung durch Übungsvorstellung, Präsentationen, Mitarbeit, Abgabe der Projektübungen.

Lehrveranstaltungen des Moduls:

6.0/4.0 VU Parallel Computing Einführung paralleles Rechnen

Programm- und Systemverifikation

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse:

- Kenntnis unterschiedlicher Spezifikationsformalismen, ihrer Semantik und ihrer Anwendungsgebiete
- Kenntnis unterschiedlicher Verifikationstools
- Verständnis grundlegender Methoden der Modellierung in Hinsicht auf Verifikationsfragen

- Beispielhafte Kenntnisse zu Zertifikation und Industriestandards in Hinsicht auf Verifikation

Kognitive und praktische Fertigkeiten:

- Praktischer Umgang mit Spezifikationsformalismen hinsichtlich ihrer Semantik und hinsichtlich Requirement Engineering
- Praktischer Umgang mit Verifikationstools
- Praktische Modellierung und Verifikation von Systemen und Interpretation der Ergebnisse

Soziale Kompetenzen, Innovationskompetenz und Kreativität:

- Verständnis für das Gefahrenpotential fehlerhafter Software und Hardware
- Verständnis für die Bedeutung formaler Methoden in der Produktentwicklung
- Anwendung theoretischer Konzepte auf angewandte Fragestellungen

Inhalt:

- Methoden der Modellierung und Spezifikation durch Logik, Automaten, Assertions, Coverage Kriterien
- Verifikationswerkzeuge, insbesondere Model Checker, Statische Analyse, Theorem-beweisen, Testen
- Praktischer Umgang mit Verifikationswerkzeugen
- Grundlagen zur Zertifizierung und zu Standards in der industriellen Validierung

Erwartete Vorkenntnisse:

Fachliche und methodische Kenntnisse: Programmkonstruktion, Modellierung, theoretische Informatik und Mathematik.

Kognitive und praktische Fertigkeiten: Geübter, fachgerechter Umgang mit Computerprogrammen und Konzepten der theoretischen Informatik und Mathematik.

Soziale Kompetenzen, Innovationskompetenz und Kreativität: Fähigkeit zur selbständigen Einarbeitung in Tools anhand schriftlicher Unterlagen.

Diese Voraussetzungen werden in folgenden Modulen vermittelt: *Algebra und Diskrete Mathematik, Theoretische Informatik und Logik, Programmkonstruktion, Modellierung*

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die in der Vorlesung vermittelten Grundlagen und Methoden werden in praktischen Übungen am Computer und auf Papier vertieft und angewandt. Die Leistungsfeststellung erfolgt durch Beispiel-Abgaben und schriftliche Tests/Prüfungen.

Lehrveranstaltungen des Moduls:

6.0/4.5 VU Programm- und Systemverifikation

Programmierparadigmen

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse: Das Modul vermittelt

- Kenntnisse der Programmiersprachen und der produktiven Programmierung aus der Sichtweise aktueller Methoden der Softwareentwicklung,
- Kenntnisse zur effizienten Entwicklung objektorientierter Programme auf Basis eines guten Verständnisses der Wiederverwendung von Programmteilen,
- Kenntnisse zur Entwicklung funktionaler Programme in einem dafür typischen Stil.

Kognitive und praktische Fertigkeiten: Durch die praktische Auseinandersetzung mit aktuellen Programmiersprachen, Programmiermethoden und Programmierwerkzeugen werden folgende Fertigkeiten vermittelt bzw. ausgebaut:

- Modellbildung und Abstraktion in der Programmierung.
- Einsatz bewährter Methoden zur Modellbildung, Lösungsfindung und Evaluation im Bereich der Programmierung.
- Umgang mit unspezifizierten und unvollständig spezifizierten Problemsituationen.
- Kritische Bewertung und Reflexion von Lösungen.

Soziale Kompetenzen, Innovationskompetenz und Kreativität:

- Der Übungsbetrieb fördert die Selbstorganisation und Eigenverantwortlichkeit sowie das Finden kreativer Problemlösungen und eigenständiger Lösungsstrategien.
- Die Arbeit in Gruppen stärkt die Teamfähigkeit.
- Die Vermittlung von Hintergrundwissen fördert die Neugierde an Themen im Bereich der Programmiersprachen und -paradigmen.

Inhalt:

- Überblick über Paradigmen der Programmierung und entsprechende Programmiersprachen
- Konzepte objektorientierter und funktionaler Programmiersprachen
- Sprachkonzepte für Generizität, Nebenläufigkeit und Modularisierung
- Produktive Verwendung dieser Konzepte in einer den zugrundeliegenden Paradigmen entsprechenden Weise

- Genaue Betrachtung der Ersetzbarkeit und anderer für die Wiederverwendung in objektorientierten Programmen bedeutender Prinzipien
- Zusammenhänge zwischen verschiedenen objektorientierten Konzepten und Prinzipien
- Ausgewählte Entwurfsmuster und objektorientierte Programmiertechniken
- Rekursive und applikative Programmierung in funktionalen Sprachen

Erwartete Vorkenntnisse:

Fachliche und methodische Kenntnisse: Objektorientierte Modellierung, grundlegende Algorithmen und Datenstrukturen

Kognitive und praktische Fertigkeiten: Praktische Programmierkenntnisse in einer objektorientierten Programmiersprache

Soziale Kompetenzen, Innovationskompetenz und Kreativität: Fähigkeit zur Zusammenarbeit in Kleingruppen.

Diese Voraussetzungen werden in folgenden Modulen vermittelt: *Algorithmen und Datenstrukturen, Modellierung, Programmkonstruktion*

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung:

- 2 Ects (je 1 Ects für Funktionale Programmierung und Objektorientierte Programmiertechniken) Vortrag und selbständiges Erlernen der Grundlagen und des theoretischen Stoffes (= Massenlehre), Beurteilung durch Prüfung oder Tests.
- 4 Ects (je 2 Ects Funktionale Programmierung und Objektorientierte Programmiertechniken) Übung in Kleingruppen zur Entwicklung praktischer Fähigkeiten, Beurteilung der Lösungen von Programmieraufgaben.

Lehrveranstaltungen des Moduls:

3.0/2.0 VU Objektorientierte Programmiertechniken

3.0/2.0 VU Funktionale Programmierung

Programmkonstruktion

Regelarbeitsaufwand: 8.8 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse: Das Modul vermittelt

- alle zur Erstellung von Programmen in einer stark typisierten objektorientierten Programmiersprache notwendigen Kenntnisse

- sowie Kenntnisse über eine systematische, konstruktive Vorgehensweise bei der Erstellung und Evaluation von Programmen.

Kognitive und praktische Fertigkeiten: Die Auseinandersetzung mit Methoden und Werkzeugen der Programmierung vermittelt

- die praktische Fähigkeit zur Konstruktion von Programmen,
- eine abstrakte und systemorientierte Denkweise in der Programmierung,
- sowie die Fähigkeit zum Einsatz einfacher formaler und informeller Methoden bei der Erstellung und Evaluation von Programmen.

Soziale Kompetenzen, Innovationskompetenz und Kreativität: Folgende Kompetenzen werden besonders gefördert:

- Selbstorganisation und Eigenverantwortlichkeit,
- Fähigkeit zur Zusammenarbeit in kleinen Teams,
- Neugierde an der Programmierung.

Inhalt:

- Ziele und Qualitätsbegriff in der Programmierung
- Abstraktes Modell (= Objekt oder abstrakte Maschine), dessen Verhalten durch ein Programm beschrieben wird
- Grundlegende Sprachkonstrukte und ihre Anwendungen
- Zusicherungen (formal und informell) und Testfälle zur Spezifikation des Programmverhaltens und zur Evaluation
- Umgang mit und Strategien zur Vermeidung von Laufzeitfehlern (Debugging, Exception Handling, Programmanalyse)
- Verwendung einer Programmierumgebung und von Programmierwerkzeugen
- Problemlösungsstrategien, Datenstrukturen und Algorithmen
- Implementierung von Listen, Stapelspeichern und Bäumen
- Rekursion in Datenstrukturen und Algorithmen
- Prinzipien der objektorientierten Programmierung (Datenabstraktion, Untertypen, Polymorphie, Vererbung)
- Verwendung von Standardbibliotheken

- Ein- und Ausgabe sowie die interne Repräsentation von Daten
- Basiswissen über Generizität und nebenläufige Programmierung
- Sicherheit in der Programmierung (Gefahrenquellen und Vermeidungsstrategien)
- Verweise auf und Beispiele in andere(n) Programmiersprachen zur Förderung des Interesses an der Programmierung

Erwartete Vorkenntnisse: Fundierte Mathematik-Kenntnisse auf AHS/BHS-Maturaniveau.

Verpflichtende Voraussetzungen: Keine.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung:

- 1.9 Ects Vortrag und selbständiges Erlernen der eher theoretischen Grundlagen (Vorlesung). Beurteilungsformen: Prüfung, Tests.
- 1.0 Ects Übung mit Aufgaben zu ausgewählten Fragestellungen zur Festigung und Reflexion wichtiger Aspekte der eher theoretischen Grundlagen. Beurteilungsformen: Abgaben, Tests.
- 5.9 Ects Übung zur Entwicklung praktischer Programmierfähigkeiten mit Betreuung in geführten Kleingruppen, wobei der Zeitrahmen flexibel gehalten wird, um Unterschiede in den Vorkenntnissen der StudentInnen auszugleichen (auf bis zu 2 Semester dehnbar). Beurteilungsformen: Lösungen von Programmieraufgaben, Tests.

Lehrveranstaltungen des Moduls:

2.9/2.0 VU Grundlagen der Programmkonstruktion

5.9/4.0 UE Programmierpraxis

Security

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele: Durch die Absolvierung dieses Moduls erhalten die Studierenden Grundkenntnisse der IT-Sicherheit. Die Studierenden lernen Aspekte der IT-Sicherheit in Projekten zu identifizieren und Maßnahmen zu setzen, um diese zu berücksichtigen.

Fachliche und methodische Kenntnisse: Das Modul vermittelt

- die theoretische Grundlagen der IT-Sicherheit,
- wichtige Sicherheitsaspekte in IT-Projekten, und
- Wissen über wichtige Best-Practice Sicherheitsmaßnahmen.

Kognitive und praktische Fertigkeiten:

- Umsetzung von wichtigen Best-Practice Sicherheitsmaßnahmen
- Verstehen der Denkweise von AngreiferInnen

Soziale Kompetenzen, Innovationskompetenz und Kreativität:

- Aufmerksamkeit für Sicherheitsaspekte in IT-Projekten
- Kommunikation der Relevanz von IT-Sicherheit und von Lösungsideen für IT-Sicherheitsprobleme

Inhalt:

- Operating systems
- Access control
- Network security
- Application security
- Database security
- Web (application) security
- Applied cryptography
- Privacy
- Security and usability
- Operations security
- Physical security
- Security architecture
- Forensics
- Exploiting vulnerabilities
- Malware
- System security
- Standards, policies, best practices
- Ethics, Compliance, Legal investigations

- Risk management
- Information security
- Business continuity, disaster recovery

Erwartete Vorkenntnisse:

Dieses Modul baut auf den Kenntnissen und Fertigkeiten folgender Module auf: *Programmkonstruktion, Security und Recht, Technische Grundlagen der Informatik*

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: VU, VO, UE

Lehrveranstaltungen des Moduls:

3.0/2.0 VU Internet Security

3.0/2.0 VU Security for Systems Engineering

Security und Recht

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse:

- Einführung in die IT-Security
- Wissen über typische Sicherheitsprobleme und deren Beseitigung
- Einführung in das IT-bezogene Technikrecht bzw. zur rechtlichen Problematik des Internet

Kognitive und praktische Fertigkeiten:

- Einfühlungsvermögen in die Sichtweise potentieller Angreifer
- Erfahrung in Angriff und Sicherung von Systemen
- Richtige Anwendung von kryptographischen Techniken
- Orientierung in der Rechtsordnung (IT-bezogenes Völkerrecht, EU-Recht und nationales Recht)
- Einsicht in die Problematik der Regulierung von Technik durch Recht
- Verständnis in Bezug auf Strukturen und Zusammenhänge im IT-bezogenen Technikrecht
- Befähigung zur Subsumtion von (einfachen) Sachverhalten unter die relevanten technikrechtlichen Vorschriften

Soziale Kompetenzen, Innovationskompetenz und Kreativität:

- Sensibilisierung für IT-Security in Forschung und Wirtschaft
- Entwicklung eines Gefühls für wissenschaftliche und wirtschaftliche Herausforderungen, Lösungsansätze und ihre Beurteilung
- Befähigung zur selbständigen, basalen Erschließung weiterer IT-rechtlicher Themengebiete
- Gesprächsfähigkeit betreffend IT-Themen aus rechtlicher Perspektive

Inhalt:

- Einführung in die Security
 - Grundlagen der Security
 - Zugriffskontrolle
 - Betriebssystemsecurity
 - Netzwerksecurity
 - Grundlagen der Kryptographie
 - Security von Anwendungsprogrammen
 - Websecurity
 - Auffinden von Schwachstellen
- Daten und Informatikrecht
 - Grundlagen zu Staat und nationalem Recht
 - Grundlagen und Grundbegriffe des Internationalen Rechts/Völkerrechts
 - Grundlagen und Grundbegriffe des EU-Rechts
 - Sonderprobleme der Regulierung von Technik (insb. IT) durch Recht
 - Grundrechte in der Informationsgesellschaft
 - Ausgewählte Gebiete bzw. Probleme des materiellen Internetrechts
 - Grundlagen der Rechtsinformationsrecherche via Internet

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: VO, VU, blended learning

Blended Learning

- Unterstützung durch ein e-learning-System (TUWEL)
- Frontalunterricht für die theoretischen Grundlagen

- Aufzeichnung/Podcasts von Vorträgen (technische Vorträge, CERIAS-Seminare)
- Praktische Arbeiten am Computer
- Unterstützung durch Tutoren

Lehrveranstaltungen des Moduls:

3.0/2.0 VU Introduction to Security

3.0/2.0 VU Daten- und Informatikrecht

Software Engineering und Projektmanagement

Regelarbeitsaufwand: 9.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse: Das Modul vermittelt Kenntnisse zur Softwareerstellung und -wartung:

- Zusammenführen der isolierten Kenntnisse und Fähigkeiten aus den relevanten vorgehenden Lehrveranstaltungen zu einer praxisnahen Gesamtsicht von der softwaretechnischen Problemstellung zur Lösung.
- Kenntnis der wichtigsten Begriffe der Softwaretechnik, der Bedeutung zentraler Konzepte im Software Engineering und Bewusstsein für die wesentlichen Projektphasen und -rollen Fähigkeit relevante Konzepte und Methoden für die einzelnen Phasen eines Software-Engineering-Projekts anzuwenden (Algorithmen, Datenstrukturen und Programmierung, Datenbanken, theoretischer Informatik)
- Kenntnis der wichtigsten Vorgehensmodelle im Software-Engineering sowie deren Unterschiede in Bezug auf konkrete Einsatzfälle
- Fähigkeit zur Anwendung eines praxisrelevantes Software-Prozessmodells (z.B. Unified Process oder Scrum)
- Kenntnis der Methoden des Software-Projektmanagements (z.B. Aufwandsschätzung und Risikomanagement)

Kognitive und praktische Fertigkeiten:

- Systemorientierte und flexible Denkweise: Auswahl, Erarbeitung und sachgerechte Anwendung von Konzepten, Modellen und Werkzeugen im Rahmen eines komplexen Software-Entwicklungsprojekts
- Methodisch fundierte Herangehensweise an Probleme, insbesondere im Umgang mit ggf. unspezifizierten Problemsituationen
- Praktische Anwendung von Techniken für Abstraktion und Modellbildung

- Hochwertige und Transparenz-schaffende Planung und Dokumentation

Soziale Kompetenzen, Innovationskompetenz und Kreativität:

- Teamfähigkeit: Problemlösung und Umsetzung in einem verteilt arbeitenden Team
- Selbstorganisation und Eigenverantwortlichkeit
- Entscheidungsverantwortung und Führungskompetenz in komplexen Projekten oder Tätigkeiten gemäß einem Rollenkonzept
- Kollaborativer Besitz und Wissensmanagement in einem mittelgroßen Team
- Eigeninitiative und Neugierde auf innovative und kreative Konzepte und Lösungsansätze
- Kenntnisse der eigenen Fähigkeiten und Grenzen
- Erfahren einer Auftraggeber-Auftragnehmer Beziehung inkl. überzeugender Präsentation

Inhalt: Folgende Inhalte werden vermittelt:

- Ingenieurdisziplin Softwaretechnik und die Bausteine eines Projektes
- Vorgehensmodelle und Rollen im Software Engineering

Anforderungsanalyse und Spezifikation - Was soll gebaut werden?

- Systementwurf und Architektur - Wie wird technisch gebaut?
- Implementierung - Wie wird codiert?
- Integration und Test - Wie wird zusammengefügt und geprüft?
- Inbetriebnahme, Rollout und Wartung
- Grundkenntnisse des Projektmanagements
- Qualitätssicherung im Kontext der Softwareentwicklung
- Rolle des Usability Engineering und Security in der Softwareentwicklung
- Fallbeispiele von Projekten zur Diskussion der Anwendung von Konzepten, Methoden und Werkzeugen

Erwartete Vorkenntnisse:

Fachliche und methodische Kenntnisse: Objektorientierte Analyse, Design und Programmierung; Grundkenntnisse aus Algorithmen und Datenstrukturen; Grundkenntnisse zu Datenbanksystemen

Diese Voraussetzungen werden in folgenden Modulen vermittelt: *Modellierung, Programmierparadigmen, Programmkonstruktion*

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Das Modul setzt sich zusammen aus einer Vorlesung, in der die theoretischen Konzepte und methodischen Grundlagen vorgestellt sowie Erfahrungen aus der praktischen Übung reflektiert werden, und einer Übung, in der ein mittelgroßes Software Engineering Projekt mit dem Ziel eines real brauchbaren Software-Prototyps und zugehöriger Dokumentation in einer Kleingruppe mit intensiver Betreuung durchgeführt wird. Intensive Betreuung der Teams in wöchentlichen Treffen mit dem Tutor; Auffrischung der Vorkenntnisse in Tutorien zu Beginn der LU. Leistungsbeurteilung durch einen Eingangstest, um die Vorkenntnisse zu überprüfen, Zwischenabgaben und -präsentationen, sowie einer praktischen und theoretischen Prüfung am Ende.

Lehrveranstaltungen des Moduls:

3.0/2.0 VO Software Engineering und Projektmanagement

6.0/4.0 PR Software Engineering und Projektmanagement

Softwareprojekt-Beobachtung und -Controlling

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse: Dieses Modul befasst sich mit der strategischen qualitätserzeugenden Supervision von SW- und IT-Projekten in den vielfältigen Anwendungsbereichen der Informatik (z.B. Betriebliche Informationssysteme, Internet-Plattformen, Versicherungs-IT, Medizinische Informatik). Typische methodische Schwachstellen in der Praxis sollen anhand von konkreten Projekt- und System-Beobachtungen identifiziert und erlebt werden. Ihre Behebung im konkreten Umfeld soll konzipiert und diskutiert werden. Ziel ist, den Studierenden die Ursachen des Abstandes zwischen SE-Theorie und konkreter Praxis verständlich zu machen und in Form konkreter Fallbeispiele die Behebung erfahrbar zu machen.

Kognitive und praktische Fertigkeiten: Das Modul vermittelt Kenntnisse über Strategien und Konzepte zur Begleitung, Bewertung und Steuerung von Software-Projekten. Es vermittelt Kenntnisse über Kriterien für erfolgreiche Projekte sowie deren Evaluierung und Beurteilung anhand etablierter Software Engineering- und Projekt-Management-Methoden.

Soziale Kompetenzen, Innovationskompetenz und Kreativität: Das Modul behandelt die Notwendigkeit effektiver Kommunikation zur Bewertung und Steuerung von Softwareentwicklungsprojekten. Es wird die Bedeutung von pragmatischem Risikomanagement, realitätsbezogenen Projektkennzahlen und interdisziplinärem Denken für den Projekterfolg vermittelt und auf die Möglichkeiten und Herausforderungen bei der Einbindung agiler Software-Entwicklungsmethoden wie z.B. Scrum eingegangen.

Inhalt:

- Basisinhalte aus Qualitätssicherung (QS) und Usability
- Fallbeispiele aus konkreten Softwareprojekten, Schwerpunkt auf Erhebungsmethoden aus QS und Usability

Erwartete Vorkenntnisse:

Diese Modul baut auf den Kenntnissen, Fertigkeiten und Kompetenzen des Moduls *Softwarequalitätssicherung* auf.

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: In Form von Vorlesungen mit Übungen werden die vertiefenden Inhalte vermittelt.

Lehrveranstaltungen des Moduls:

6.0/4.0 VU Softwareprojekt-Beobachtung und -Controlling

Softwarequalitätssicherung

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse:

- Motivation und Ziele der Softwarequalitätssicherung
- Qualitätskostenmodelle und Kostenoptimierung
- Organisatorische Qualitätssicherung
- Statische und dynamische Methoden der Qualitätssicherung
- Qualitätsmanagementstandards
- Formale Grundlagen für Methoden der Qualitätssicherung
- Methoden zur Feststellung und Verbesserung gewünschter Qualitätsmerkmale in Softwaresystemen
- Methoden zur Sicherstellung und Verbesserung der Qualität von Produkten.

- Methoden um Personen zu führen, deren Fähigkeiten gezielt einzusetzen und weiterzuentwickeln.
- Methoden zur Definition und schrittweisen Verbesserung von Prozessen für Analyse, Entwurf, Implementierung, Test, Inbetriebnahme und Wartung von Softwaresystemen.

Kognitive und praktische Fertigkeiten:

- Systemorientierte und flexible Denkweise: Auswahl, Erarbeitung und sachgerechte Anwendung von Konzepten, Modellen und Werkzeugen zur Qualitätssicherung im Rahmen eines komplexen Software-Entwicklungsprojekts
- Methodisch fundierte Herangehensweise an Probleme, insbesondere im Umgang mit ggf. unspezifizierten Problemsituationen
- Selektion der passenden QS-Methoden in einem Entwicklungsprojekt
- Praktische Anwendung von Techniken für Abstraktion und Modellbildung
- Auswahl und Anwendung geeigneter Metriken zur Qualitätsbestimmung
- Anwendung von statischen und dynamischen Methoden zur Softwarequalitätssicherung z.B. Softwaretests
- Hochwertige und Transparenz-schaffende Planung und Dokumentation

Soziale Kompetenzen, Innovationskompetenz und Kreativität:

- Präsentation und Diskussion eines Qualitätsplans
- Selbstorganisation und Eigenverantwortlichkeit
- Eigeninitiative und Neugierde auf innovative und kreative Konzepte und Lösungsansätze
- Kenntnisse der eigenen Fähigkeiten und Grenzen

Inhalt: Inhalte des Vorlesungsteils:

- Grundlagen der Software-Qualitätssicherung
- Statische Qualitätssicherung
- Dynamische Qualitätssicherung
- Organisatorische Qualitätssicherung
- Qualitätssicherungs-Standards

Folgende praktische Inhalte werden im Rahmen der Laborübung vermittelt:

- Review von Designs/Modellen
- Kollaborative Code-Inspektionen
- Statische Code Analyse / Antipattern Analyse
- Test-Driven Development
- Testplanerstellung inkl. Ableiten effizienter und effektiver Testfälle
- Testautomatisierung Whitebox
- Testautomatisierung Blackbox

Erwartete Vorkenntnisse:

Fachliche und methodische Kenntnisse: Folgende Vorkenntnisse (Anwendung der Methoden im Rahmen von praxisrelevanten Aufgabenstellungen) sind für das erfolgreiche Absolvieren notwendig:

- Mathematik und Statistik
- Objektorientierte Analyse, Design und Programmierung
- Grundlagen der Unified Modeling Language (UML)
- Beherrschung einer praxisrelevanten Programmiersprache und -werkzeuge (z.B. Java oder C++)
- Umgang mit Integrierten Entwicklungsumgebungen, Build Management und Quellcodeverwaltung
- Kenntnis von wesentlichen Architekturstilen und Design-Patterns
- Grundkenntnisse zu Datenbanksystemen

Diese Vorkenntnisse werden in folgenden Modulen vermittelt: *Modellierung, Programmkonstruktion, Software Engineering und Projektmanagement*

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die Lehrveranstaltung setzt sich zusammen aus einem Vorlesungsteil, in dem die theoretischen Konzepte und Lösungsansätze vorgestellt werden, und einem Übungsteil, in dem praktische Beispiele aus den Bereichen Reviews und Testen am Computer umgesetzt werden. Im Rahmen eines realitätsnahen mittelgroßen Projektes sollen typische Aufgaben der Softwarequalitätssicherung gelöst werden. Intensiver Einsatz von entsprechenden Werkzeugen z.B. Testautomatisierung zur Umsetzung der QS-Konzepte und -Methoden. Intensive

Betreuung durch Tutoren in wöchentlichen Treffen. Präsentationen der Studierenden zu Lösungsansätzen und Fortschritten der Lösungserarbeitung.

Lehrveranstaltungen des Moduls:

6.0/4.0 VU Software-Qualitätssicherung

Statistik und Wahrscheinlichkeitstheorie

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele: Vermittlung der statistischen Denk- und Arbeitsweise

Fachliche und methodische Kenntnisse: Grundlagen der Wahrscheinlichkeitstheorie; Kenntnisse von statistischer Schätzung und statistischem Testen; Kenntnisse wichtiger statistischer Methoden

Kognitive und praktische Fertigkeiten: Anwendung von statistischen Methodiken auf konkrete Problemstellungen; Kenntnisse im Umgang mit statistischer Software

Soziale Kompetenzen, Innovationskompetenz und Kreativität: Umsetzung von konkreten Aufgaben in statistische Problemstellungen; Lösung statistischer Problemstellungen sowohl formal als auch mit dem Computer

Inhalt: Dieses Modul vermittelt im Einzelnen folgende Themen: Beschreibende Statistik, Grundlagen der Wahrscheinlichkeitstheorie, Elementare Informationstheorie, Zufallsvariablen und Verteilungen, Punkt- und Intervallschätzungen, Tests von Hypothesen, Varianzanalyse, Regression, Korrelation, Zählstatistik.

Erwartete Vorkenntnisse: Grundkenntnisse der Analysis und Algebra.

Diese Voraussetzungen werden in folgenden Modulen vermittelt: *Algebra und Diskrete Mathematik, Analysis*

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die Lehrveranstaltung Statistik und Wahrscheinlichkeitstheorie besteht aus einem Vorlesungsteil und einem Übungsteil. Die beschriebenen Inhalte und Konzepte werden im Rahmen der Vorlesungseinheit erläutert. Der Übungsteil besteht aus einem Teil, bei dem Beispiele analytisch gelöst werden, und einem Teil, bei dem praktische Problemstellungen mit Hilfe statistischer Software gelöst werden. Diese Veranstaltungen sollen sowohl im Winter- als auch im Sommersemester angeboten werden.

Lehrveranstaltungen des Moduls:

3.0/2.0 VO Statistik und Wahrscheinlichkeitstheorie

3.0/2.0 UE Statistik und Wahrscheinlichkeitstheorie

Statistische Datenanalyse

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele: Vermittlung einer datenorientierten und explorativen Analyse von Daten
Fachliche und methodische Kenntnisse:

- Grundlagen der statistischen Datenanalyse
- Detaillierte Kenntnisse und Hintergrundwissen über statistische Methodik

Kognitive und praktische Fertigkeiten:

- Anwendung von statistischer Methodik und Software auf konkrete Problemstellungen
- Programmentwicklung mit statistischer Software
- Computerorientierte Lösung von statistischen Problemen

Soziale Kompetenzen, Innovationskompetenz und Kreativität:

- Datenorientierte Lösung von statistischen Problemstellungen
- Lösungen von Problemen mit open-source Software

Inhalt: Stichprobendesign, Planung der statistischen Datenerhebung, Elemente der explorativen Datenanalyse, Grundbegriffe parametrischer/nichtparametrischer und robuster Verfahren, lineare Modelle, Einführung in multivariate statistische Methoden, Einführung in die Zeitreihenanalyse, effiziente Programmierung in der statistischen Softwareumgebung R, Datenmanipulation, statistische Graphiken, graphische Systeme in R, dynamische reports mit statistischer Software, Testen mittels statistischer Simulation.

Erwartete Vorkenntnisse: Grundlegende Kenntnisse der Statistik

Diese Voraussetzungen werden in im Modul *Statistik und Wahrscheinlichkeitstheorie* vermittelt.

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die Lehrveranstaltung Datenanalyse vermittelt sowohl den theoretischen Zugang zu den Methoden, demonstriert aber auch die praktische Lösung mittels Statistik-Software. Die Studierenden sollen selbständig Problemstellungen mit dem Computer lösen.

Die Lehrveranstaltung Statistical Computing vermittelt einen computerorientierten Zugang zur Statistik. Ziel ist einerseits detaillierte Kenntnisse in der State-of-the-art Software R zu vermitteln, als auch statistische Probleme mittels statistischer Simulation zu lösen. Lösungen sollen durch eigenständiges Programmieren erarbeitet werden.

Lehrveranstaltungen des Moduls:

3.0/2.0 VU Datenanalyse

3.0/2.0 VU Statistical Computing

Studieneingangsgespräch

Regelarbeitsaufwand: 0.2 Ects

Bildungsziele: Ziel des Studieneingangsgesprächs ist es, Studieninteressierte zu einer expliziten Reflexion über ihre Studienmotivation anzuregen und ihnen die Möglichkeit zu bieten, durch ein Gespräch mit in Lehre und Forschung ausgewiesenen Experten und Expertinnen die Gründe für die Studienwahl und Erwartungen an das Studium zu überprüfen.

Inhalt: Die angehenden Studierenden verfassen eigenständig ein Motivationsschreiben und führen ein Gespräch mit Angehörigen der Fakultät über ihre Motivation und Erwartungen.

Erwartete Vorkenntnisse: Keine.

Verpflichtende Voraussetzungen: Keine.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die Lehrveranstaltung wird durch *mit Erfolg teilgenommen* beurteilt, wenn ein eigenständiges Motivationsschreiben verfasst und das Studieneingangsgespräch geführt wurde.

Lehrveranstaltungen des Moduls:

0.2/1.0 UE Studieneingangsgespräch

Technische Grundlagen der Informatik

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse: Grundkenntnisse über Aufbau und Funktionsweise von Computersystemen

Kognitive und praktische Fertigkeiten: Modellbildung und Abstraktion, methodisch fundierte Herangehensweise an Probleme bei der Erstellung und Evaluation von Programmen.

Soziale Kompetenzen, Innovationskompetenz und Kreativität: Selbstorganisation und Eigenverantwortlichkeit

Inhalt:

- Kenntnisse zur Darstellung von Zahlen in Computern
- Grundlagen der Boole'schen Algebra und Minimierungsverfahren
- Grundlagen digitaler Schaltungstechnik
- Gatterschaltungen (Addierer, Codierer, Multiplexer, ...)
- Schaltnetze mit programmierbaren Bausteinen (ROM, PROM/EPROM, PLA, PAL)

- Speicherglieder (RS, JK, D) und Speicher (statisch, dynamisch)
- Synthese und Analyse von Schaltwerken
- Prozessor
- Adressierungsarten, Befehlssatz, RISC/CISC und Pipelining
- Speicherverwaltung
- Ein/Ausgabe und Peripheriegeräte
- Systemsoftware (Kurzüberblick)

Erwartete Vorkenntnisse: Fundierte Mathematik-Kenntnisse auf AHS/BHS-Maturaniveau.

Verpflichtende Voraussetzungen: Keine.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Frontalvortrag mit Prüfung. In Abhängigkeit der Bedeckbarkeit Übung in Großgruppen (Papier und Tafel) zur Festigung des Lehrstoffes. Beurteilung durch Tests.

Lehrveranstaltungen des Moduls:

6.0/4.0 VU Technische Grundlagen der Informatik

Theoretische Informatik und Logik

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele: Dieses Modul befasst sich mit den theoretischen und logischen Grundlagen der Informatik.

Fachliche und methodische Kenntnisse: Fundamentale Konzepte und Resultate der Mathematischen Logik, Automaten und formalen Sprachen, Berechenbarkeit und Komplexität sowie der formalen Semantik von Programmiersprachen.

Kognitive und praktische Fertigkeiten: Die Studierenden erwerben die Fähigkeit, formale Beschreibungen lesen und verstehen und Konzepte formal-mathematisch beschreiben zu können. Weiters lernen sie, die Struktur von Beweisen und Argumentationen zu verstehen und selber solche zu führen.

Soziale Kompetenzen, Innovationskompetenz und Kreativität: Selbständiges Lösen von Problemen.

Inhalt:

- Mathematische Logik: Aussagenlogik, Prädikatenlogik, elementare Modallogiken wie LTL, Kripkemodelle, Kalkülbegriff, logische Struktur formaler Beweise
- Automatentheorie: endliche Automaten, Büchautomaten, Transducer, Operationen auf Automaten

- Formale Sprachen: Chomsky Hierarchie
- Berechenbarkeit und Komplexität: universelle Berechenbarkeit, Unentscheidbarkeit, NP-Vollständigkeit
- Grundlagen der operationalen und axiomatischen Semantik von Programmiersprachen
- Grundlagen von Prozessalgebren und Concurrency (CSP, CCS)

Erwartete Vorkenntnisse:

Fachliche und methodische Kenntnisse: Automaten, reguläre Ausdrücke, Grammatiken sowie Aussagen- und Prädikatenlogik als Spezifikationssprachen, Syntax und Semantik, Modellbegriff.

Diese Voraussetzungen werden in der Lehrveranstaltung *Formale Modellierung* des Moduls *Modellierung* vermittelt.

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die Inhalte werden in einem Vorlesungsteil vorgestellt und in begleitenden Übungen von den Studierenden erarbeitet. Die Übungsaufgaben können zeitlich und örtlich weitgehend ungebunden einzeln oder in Gruppen gelöst werden. Die Lösungen werden bei regelmäßigen Treffen mit Lehrenden und TutorInnen besprochen und korrigiert. Die Beurteilung erfolgt auf Basis schriftlicher Tests und der kontinuierlich in den Übungen erbrachten Leistungen. Der Übungsbetrieb und die Tests können computerunterstützt durchgeführt werden.

Lehrveranstaltungen des Moduls:

6.0/4.0 VU Theoretische Informatik und Logik

Übersetzerbau

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse:

- Theoretische Grundlagen des Übersetzerbaus

Kognitive und praktische Fertigkeiten: Die Auseinandersetzung mit Methoden und Werkzeugen des Übersetzerbaus vermittelt

- die praktische Fähigkeit zur Assemblerprogrammierung und
- die praktische Fähigkeit zur Konstruktion von Parsern und Übersetzern

Soziale Kompetenzen, Innovationskompetenz und Kreativität:

- Neugierde am Übersetzerbau

Inhalt:

- Grundlagen von Compilern und Interpretern, Struktur von Übersetzern
- Computerarchitektur und Grundlagen
- Lexikalische Analyse (reguläre Definition, endlicher Automat)
- Syntax-Analyse (Top-Down, Bottom-Up)
- Syntaxgesteuerte Übersetzung (Attributierte Grammatik)
- Semantische Analyse, Zwischencode (Symboltabelle)
- Zwischencode
- Codeerzeugung (Befehlsauswahl, Befehlsanordnung, Registerbelegung)
- Laufzeitsystem (Stackverwaltung, Heapverwaltung)
- Optimierungen (Programmanalysen, skalare Optimierungen, Schleifenoptimierungen)
- Übersetzung objektorientierter Konzepte (Klassendarstellung und Methodenauf-ruf, Typüberprüfung, Analysen)

Erwartete Vorkenntnisse:

- theoretische Grundlagen der Informatik
- alle zur Erstellung von Programmen notwendigen Kenntnisse
- die praktische Fähigkeit zur Konstruktion von Programmen

Diese Voraussetzungen werden in folgenden Modulen vermittelt: *Programmkonstruktion, Modellierung, Algorithmen und Datenstrukturen, Programmierparadigmen*

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Vortrag und selbständiges Erlernen der eher theoretischen Grundlagen. Laborübung in geführten Kleingruppen zur Entwicklung praktischer Übersetzerentwicklungsfähigkeiten. Die Beurteilung erfolgt durch Prüfung oder Tests und die Beurteilung der Lösungen von Programmieraufgaben plus Abschlussgespräch.

Lehrveranstaltungen des Moduls:

6.0/4.0 VU Übersetzerbau

Usability Engineering and Mobile Interaction

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse: Dieses Modul befasst sich mit der User-Research getriebenen Konzeption, Gestaltung und Evaluierung von Benutzerschnittstellen. Das Modul gliedert sich in zwei große Teilgebiete: Im ersten Teil werden die grundlegenden Konzepte von Usability Engineering gelehrt. Anhand von praxisnahen Beispielen sollen Studierende den Einsatz von Usability Engineering erlernen. Die gelehrteten Methoden decken den gesamten Design Prozess von Requirements Engineering (z.B. Kontextuelle Interviews), Prototyping bis hin zum Testen von Systemen (z.B. Usability Test, Heuristische Evaluierung) ab. Der zweite Teil dieses Moduls ist den Methoden der Mobile Interaction Research gewidmet, mit besonderem Fokus auf aktuellen Entwicklungen und Trends. Aufbauend auf den Grundlagen des Usability Engineerings werden Besonderheiten und Spezifika sowohl im Design als auch in der Evaluierung von mobilen Anwendungen hervorgehoben.

Kognitive und praktische Fertigkeiten: Das Modul vermittelt Kenntnisse über Qualitätskriterien für gute Usability sowie deren Evaluierung und Beurteilung anhand etablierter Usability Engineering Methoden und zeigt aktuelle Entwicklungen und zukünftige Trends im Bereich der Mobile Interaction auf.

Soziale Kompetenzen, Innovationskompetenz und Kreativität: Das Modul vermittelt die Bedeutung von Usability Engineering für den Erfolg von Softwareentwicklungsprojekten und geht auf die Möglichkeiten und Herausforderungen der Einbindung von Usability Engineering Methoden in Software Engineering Prozessen ein.

Inhalt: Usability Engineering:

- Einführung in Usability Engineering
- Qualitätskriterien für Usability Engineering und deren Messung und Beurteilung
- Usability Engineering Lifecycle
- Methoden des Usability Engineerings in Anlehnung an die Phasen des Human Centered Design Prozesses: Kontextanalyse, Requirementsanalyse, Design and Prototyping, Evaluierung
- Praktische Anwendung der vorgestellten Methoden in einem Übungsteil

Pilots in Mobile Interaction: User-centered Interaction Research and Evaluation:

- Einführung in User-centered Interaction Research
- Quality of Experience Methods and Applications
- Perceptual Quality for Mediated Interaction

- Cognitive User Interfaces
- Audio-Visual Speech Synthesis
- Advanced Mobile Spatial Interaction
- Rapid Prototyping for Future Mobile Interactions
- Case Study of Mobile Interfaces in Large IT Projects

Verpflichtende Voraussetzungen: Studiengang- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: In Form von Vorlesungen mit Übungen werden die vertiefenden Inhalte vermittelt. In der Vorlesung werden theoretische Grundlagen vermittelt. Die in der Vorlesung vorgestellten Methoden sind in einem praktischen Übungsteil umzusetzen.

Lehrveranstaltungen des Moduls:

3.0/2.0 VU Usability Engineering

3.0/2.0 VU Pilots in Mobile Interaction: User-centered Interaction Research and Evaluation

Verteilte Systeme

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse:

- Verteilte Systeme, Internetcomputing
- Anforderungen und Design-Möglichkeiten komplexer, verteilter Systeme verstehen
- Grundlegende Methoden und Algorithmen Verteilter Systeme verstehen, sowie deren Vor- und Nachteile und Einsatzmöglichkeiten kennen
- Paradigmen und Konzepte aktueller Technologien und Werkzeuge für Verteilte Systeme verstehen und anwenden können
- Anwendungsgrenzen (v.a. asynchroner) Verteilter Systeme kennen und verstehen

Kognitive und praktische Fertigkeiten: Die Auseinandersetzung mit Methoden und Werkzeugen der Programmierung vermittelt

- Abstraktion
- methodisch fundierte Herangehensweise an Probleme
- kritische Bewertung und Reflexion von Lösungen

- Programmatische Umsetzung der Konzepte Verteilter Systeme mit aktuellen Technologien in Form einfacher, verteilter Anwendungen

Soziale Kompetenzen, Innovationskompetenz und Kreativität: Folgende Kompetenzen werden besonders gefördert:

- Selbstorganisation und Eigenverantwortlichkeit,
- Eigeninitiative und Neugierde
- Finden kreativer Problemlösungen

Inhalt:

- Grundlagen und Konzepte
- Communication und Middleware
- Operating System Support
- Naming and Discovery
- Clocks and Agreement
- Consistency and Replication
- Dependability and Fault Tolerance
- Security
- Technology Overview

Erwartete Vorkenntnisse:

Dieses Modul baut auf den Kenntnissen, Fertigkeiten und Kompetenzen folgender Module auf: *Algorithmen und Datenstrukturen, Programmkonstruktion*

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Blended Learning:

- Den Studierenden wird empfohlen vor der jeweiligen Vorlesung die auf der LVA Homepage angegebenen Buchseiten zu lesen und die Fragen aus dem Fragenkatalog zu beantworten.
- Im Rahmen der Vorlesung wird die Theorie erläutert und Querverbindungen hergestellt. Es besteht die Möglichkeit komplexe Sachverhalte interaktiv (durch Fragen der Studierenden) zu erarbeiten.

- Im Rahmen der parallel laufenden Laborübungen werden ausgewählte Themen der Lehrveranstaltung durch kleine Programmieraufgaben vertieft

Lehrveranstaltungen des Moduls:

3.0/2.0 VO Verteilte Systeme

3.0/2.0 UE Verteilte Systeme

Vertrags-, Daten- und Informatikrecht

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele: Eine AbsolventIn des Moduls ist dazu befähigt, konkrete Probleme des materiellen Internetrechts als solche zu erkennen, selbst zumindest grundsätzlich zu beurteilen sowie mit Jurist/inn/en bei der Lösung der Rechtsprobleme effektiv und kritisch auf interdisziplinärer Ebene zusammenzuarbeiten sowie grundlegende Probleme des (privatrechtlichen) Vertrags- und Haftungsrechts selbständig beurteilen und lösen zu können.

Fachliche und methodische Kenntnisse: Das Modul vermittelt

- Rechtsgrundlagen jeweils aktueller Probleme des materiellen Internetrechts
- eine Einführung in das (privatrechtliche) Vertrags- und Haftungsrecht

Kognitive und praktische Fertigkeiten: Das Modul vermittelt

- Sensibilität für konkrete rechtliche Aspekte von IT
- Subsumtion realer Sachverhalte unter die einschlägigen Rechtsvorschriften
- Anwendung juristischer Interpretationsregeln
- Rechtsinformationsrecherche via Internet

Soziale Kompetenzen, Innovationskompetenz und Kreativität: Das Modul vermittelt interdisziplinäre Gesprächsfähigkeit betreffend Fragen des IT-Rechts bzw. des Vertrags- und Haftungsrechts.

Inhalt:

- Daten- und Informatikrecht
 - Präsentation aktueller Fallkonstellationen mit Bezug zum Internetrecht
 - Anleitung zur selbständigen Problemanalyse und Rechtsinformationsrecherche
 - Ausarbeitung der rechtlichen Beurteilung konkreter Fälle bzw. Rechtsfragen
- Vertrags- und Haftungsrecht
 - Ordnungsfunktion des Privatrechts

- Grundlagen des Vertragsrechts
- Grundlagen des Haftungsrechts

Erwartete Vorkenntnisse:

Fachliche und methodische Kenntnisse: Grundlagen des Daten- und Informatikrechts

Kognitive und praktische Fertigkeiten: Bereitschaft zur Auseinandersetzung mit rechtlichen Konzepten und Methoden

Soziale Kompetenzen, Innovationskompetenz und Kreativität: Befähigung zur selbständigen Recherche und Problemlösung

Diese Voraussetzungen werden im Modul *Security und Recht* vermittelt.

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung:

- Frontalvortrag
- Inhaltliche Kickoffs
- Blended Learning (TUWEL-Kurs)
- Rechtsinformationsrecherche via Internet
- Praktische Falllösung online (TUWEL) bzw. konventionell (Papier)
- Schriftliche Prüfung

Lehrveranstaltungen des Moduls:

3.0/2.0 UE Daten- und Informatikrecht

3.0/2.0 VU Vertrags- und Haftungsrecht

Wissensrepräsentation

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse:

- Fundierte Kenntnisse in der Theorie der Wissensrepräsentation und der Wissensverarbeitung
- Fundamentale Konzepte, um Wissensrepräsentationsmechanismen erfolgreich anzuwenden

Kognitive und praktische Fertigkeiten:

- Fähigkeit zur formalen Analyse der eingesetzten Techniken und Methoden
- Fähigkeit, Methoden und Techniken für ein vorgegebenes Repräsentationsproblem auszuwählen
- Fähigkeit zur Analyse einfacher Aufgabenstellungen sowie die Umsetzung von Lösungen in eine geeignete Form der Wissensrepräsentation mit dazugehörigem Verarbeitungsmechanismus
- Kritische Bewertung und Reflexion von Lösungen
- Präsentation von Lösungen

Soziale Kompetenzen, Innovationskompetenz und Kreativität:

- Selbstorganisation, Eigenverantwortlichkeit
- Eigeninitiative und Neugierde
- Umgang mit formalen Beschreibungen und darauf aufsetzenden Verarbeitungsmechanismen

Inhalt:

- Komplexitätstheorie (hier insbesondere polynomielle Hierarchie inkl. prototypische Problemklassen)
- Klassische Logik zur Wissensrepräsentation (inkl. der Probleme und Limitierungen)
- Nichtmonotones Schließen
- Parakonsistentes Schließen
- Wissensrevision (belief revision)
- Logikbasierte Abduktion
- Diagnose
- Quantifizierte Boolesche Formeln
- Komplexitätsanalyse der besprochenen Verfahren

Erwartete Vorkenntnisse:

- Logikkenntnisse (Aussagen-, Prädikatenlogik, Kalküle)
- Kenntnisse in Grundlagen intelligenter Systeme

- Fähigkeit, einfache Beweise zu führen

Diese Voraussetzungen werden in folgenden Modulen vermittelt: *Algebra und Diskrete Mathematik, Modellierung, Theoretische Informatik und Logik, Grundlagen intelligenter Systeme*

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung:

- Präsentation der Lehrinhalte in einem Vorlesungsteil (Frontalvortrag)
- Selbständiges Ausarbeiten von Aufgabenstellungen durch Studierende
- Präsentation der Lösungen (inkl. der benötigten Theorie)

Leistungsbeurteilung:

- Schriftliche Prüfung und
- Bewertung der Ausarbeitungen (inkl. der Präsentation)

Lehrveranstaltungen des Moduls:

3.0/2.0 VO Logik für Wissensrepräsentation

3.0/2.0 UE Logik für Wissensrepräsentation

Zuverlässige Echtzeitsysteme

Regelarbeitsaufwand: 6.0 Ects

Bildungsziele:

Fachliche und methodische Kenntnisse: Fundierte Kenntnisse über die Konstruktion und Modellierung von zuverlässigen Systemen, die strikten zeitlichen Vorgaben gehorchen müssen. Die inkludiert die Kenntnisse von Fehlerarten, Fehlermodellen, Fehlererkennung, Fehleranalyse, Redundanzverfahren, Zuverlässigkeitsmodellierung, Verständnis und Arbeiten mit zeitabhängiger Information, Uhrensynchronisation, Echtzeitscheduling und Echtzeitkommunikation.

Kognitive und praktische Fertigkeiten: Fähigkeit der Einschätzung von Fehler- und Ausfallrisiken, Kenntnis von Verfahren zur Erhöhung der Zuverlässigkeit von Computersystemen, Analyse von zeitlichen Anforderungen an Computersysteme, Modellierung und Konstruktion von Computersystemen mit Echtzeitanforderungen, Tool-unterstützte Fehlermodellierung (inkl. Warteschlangenmodell-Simulation) von Computersystemen.

Soziale Kompetenzen, Innovationskompetenz und Kreativität: Fähigkeit zur Risikoabschätzung und zur Konstruktion sicherer Computersysteme; Erhöhung der Zuverlässigkeit von Echtzeitapplikationen; kritische Analyse von Systemvarianten und Dokumentation der Zuverlässigkeit von Systemen.

Inhalt:

- Grundlagen: Zuverlässigkeit, Wartbarkeit, Verfügbarkeit, MTTF
- Quantitative Analysen: Blockdiagramme, Fehlerbäume, Markov-Prozesse
- Sicherheit, Fehlermodelle, Wartung, Alterungsfehler, Entwurfsfehler
- Fehlertolerante Computersysteme: Redundanz, Fehlerlatenz, Voting, Recovery Blocks, N-Version-Programming, Synchronisation
- Fallstudien von zuverlässigen bzw. fehlertoleranten Systemen
- Fehler und Zuverlässigkeitsmodellierung/analyse mit Tools
- Grundlagen Echtzeitsysteme, Zeitabhängigkeit von Information, logische und temporale Ordnung
- Modellbildung von Echtzeitsystemen: Zustand und Ereignis, Komponenten, Interfaces, Echtzeitinformation
- Echtzeitkommunikation, Kommunikationsprotokolle für Echtzeitsysteme
- Uhrensynchronisation
- Fehlertoleranz in Echtzeitsystemen
- Echtzeitbetriebssysteme: Taskstruktur, Ressourcenmanagement, I/O, Scheduling, Worst-Case Zeitanalyse von Tasks
- Energieverbrauch und Energiemanagement in Echtzeitsystemen
- Design von Echtzeitsystemen: Architekturmodelle, Composability, Designprinzipien, Zertifizierung

Erwartete Vorkenntnisse:

Fachliche und methodische Kenntnisse: Kenntnisse in Boole'scher Algebra, theoretischer Informatik und Logik, Wahrscheinlichkeitsrechnung, Beschreibung und Modellierung stochastischer Prozesse, Aufbau und Funktionsweise von Microcomputern, Betriebssystemen und Netzwerken.

Kognitive und praktische Fertigkeiten: Wahrscheinlichkeitsrechnung und Modellierung, Systematisches Arbeiten mit Softwaretools, Abstraktionsvermögen.

Soziale Kompetenzen, Innovationskompetenz und Kreativität: Analyse komplexer Zusammenhänge und Wechselwirkungen, Problemlösung im Team.

Diese Voraussetzungen werden in folgenden Modulen vermittelt: *Technische Grundlagen der Informatik, Theoretische Informatik und Logik, Statistik und Wahrscheinlichkeitstheorie, Microcontroller und Betriebssysteme*

Verpflichtende Voraussetzungen: Studieneingangs- und Orientierungsphase.

Angewandte Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die Grundlagen und theoretischen Inhalte werden im Vorlesungsteil vermittelt. Praktische Fertigkeiten der Fehler- und Zuverlässigkeitsmodellierung werden in einem Übungsteil erworben, in dem die Studierenden Softwaretools verwenden und Ergebnisse in Form von Laborberichten dokumentieren.

Lehrveranstaltungen des Moduls:

2.0/2.0 VO Echtzeitsysteme

4.0/3.0 VU Dependable Systems

B. Lehrveranstaltungstypen

EX: Exkursionen sind Lehrveranstaltungen, die außerhalb des Studienortes stattfinden. Sie dienen der Vertiefung von Lehrinhalten im jeweiligen lokalen Kontext.

LU: Laborübungen sind Lehrveranstaltungen, in denen Studierende in Gruppen unter Anleitung von Betreuerinnen und Betreuern experimentelle Aufgaben lösen, um den Umgang mit Geräten und Materialien sowie die experimentelle Methodik des Faches zu lernen. Die experimentellen Einrichtungen und Arbeitsplätze werden zur Verfügung gestellt.

PR: Projekte sind Lehrveranstaltungen, in denen das Verständnis von Teilgebieten eines Faches durch die Lösung von konkreten experimentellen, numerischen, theoretischen oder künstlerischen Aufgaben vertieft und ergänzt wird. Projekte orientieren sich an den praktisch-beruflichen oder wissenschaftlichen Zielen des Studiums und ergänzen die Berufsvorbildung bzw. wissenschaftliche Ausbildung.

SE: Seminare sind Lehrveranstaltungen, bei denen sich Studierende mit einem gestellten Thema oder Projekt auseinandersetzen und dieses mit wissenschaftlichen Methoden bearbeiten, wobei eine Reflexion über die Problemlösung sowie ein wissenschaftlicher Diskurs gefordert werden.

UE: Übungen sind Lehrveranstaltungen, in denen die Studierenden das Verständnis des Stoffes der zugehörigen Vorlesung durch Anwendung auf konkrete Aufgaben und durch Diskussion vertiefen. Entsprechende Aufgaben sind durch die Studierenden einzeln oder in Gruppenarbeit unter fachlicher Anleitung und Betreuung durch die Lehrenden (Universitätslehrerinnen und -lehrer sowie Tutorinnen und Tutoren) zu lösen. Übungen können auch mit Computerunterstützung durchgeführt werden.

VO: Vorlesungen sind Lehrveranstaltungen, in denen die Inhalte und Methoden eines Faches unter besonderer Berücksichtigung seiner spezifischen Fragestellungen, Begriffsbildungen und Lösungsansätze vorgetragen werden. Bei Vorlesungen herrscht keine Anwesenheitspflicht.

VU: Vorlesungen mit integrierter Übung vereinen die Charakteristika der Lehrveranstaltungstypen VO und UE in einer einzigen Lehrveranstaltung.

C. Zusammenfassung aller verpflichtenden Voraussetzungen

Die Absolvierung des Moduls *Studieneingangsgespräch* (durch Abgabe des Motivations-schreibens und aktive Teilnahme am Gespräch) bildet die Voraussetzung für alle anderen Module des Studiums.

Die positiv absolvierte Studieneingangs- und Orientierungsphase (Abschnitt 7) ist Voraussetzung für die Absolvierung aller in diesem Studienplan angeführten Module (inklusive der Bachelorarbeit) ausgenommen die Module bzw. Lehrveranstaltungen

Algebra und Diskrete Mathematik (9.0 Ects)
Algorithmen und Datenstrukturen (9.0 Ects)
Analysis (6.0 Ects)
Einführung in Visual Computing (6.0 Ects)
Grundlagen der Human Computer Interaction (6.0 Ects)
Modellierung (9.0 Ects)
Programmkonstruktion (8.8 Ects)
Studieneingangsgespräch (0.2 Ects)
Technische Grundlagen der Informatik (6.0 Ects)

D. Semestereinteilung der Lehrveranstaltungen

Die in der nachfolgenden Semestereinteilung mit Stern markierten Lehrveranstaltungen setzen eine positiv absolvierte Studieneingangs- und Orientierungsphase (= Lehrveranstaltungen des ersten Semesters) voraus.

1. Semester (WS)

4.0/4.0 VO Algebra und Diskrete Mathematik für Informatik und Wirtschaftsinformatik
5.0/2.0 UE Algebra und Diskrete Mathematik für Informatik und Wirtschaftsinformatik
3.0/2.0 VU Datenmodellierung
3.0/2.0 VU Formale Modellierung
2.9/2.0 VU Grundlagen der Programmkonstruktion
5.9/4.0 UE Programmierpraxis
0.2/1.0 UE Studieneingangsgespräch
6.0/4.0 VU Technische Grundlagen der Informatik

2. Semester (SS)

6.0/4.0 VU Algorithmen und Datenstrukturen 1
3.0/2.0 VU Algorithmen und Datenstrukturen 2
2.0/2.0 VO Analysis für Informatik und Wirtschaftsinformatik
4.0/2.0 UE Analysis für Informatik und Wirtschaftsinformatik
3.0/2.0 VU Basics of Human Computer Interaction
6.0/5.0 VU Einführung in Visual Computing
3.0/2.0 VU Gesellschaftliche Spannungsfelder der Informatik
3.0/2.0 VU Objektorientierte Modellierung

3. Semester (WS)

- * 2.0/2.0 VO Betriebssysteme
- * 4.0/2.0 UE Betriebssysteme
- * 6.0/4.0 VU Datenbanksysteme
- * 3.0/2.0 VU Daten- und Informatikrecht
- * 3.0/2.0 VU Funktionale Programmierung
- * 3.0/2.0 VU Introduction to Security
- * 3.0/2.0 VU Objektorientierte Programmieretechniken
- * 3.0/2.0 VO Statistik und Wahrscheinlichkeitstheorie
- * 3.0/2.0 UE Statistik und Wahrscheinlichkeitstheorie

4. Semester (SS)

- * 3.0/2.0 VU Einführung in die Künstliche Intelligenz
- * 3.0/2.0 VO Software Engineering und Projektmanagement
- * 6.0/4.0 PR Software Engineering und Projektmanagement
- * 6.0/4.0 VU Theoretische Informatik und Logik

5. Semester (WS)

- * 5.0/3.0 VU Einführung in wissensbasierte Systeme
- * 3.0/2.0 VU Gesellschaftswissenschaftliche Grundlagen der Informatik
- * 3.0/2.0 VU Interface and Interaction Design
- * 3.0/2.0 VO Verteilte Systeme
- * 3.0/2.0 UE Verteilte Systeme
- * 3.0/2.0 SE Wissenschaftliches Arbeiten

6. Semester (SS)

- *10.0/5.0 PR Bachelorarbeit für Informatik und Wirtschaftsinformatik

E. Semestereinteilung für schiefeinsteigende Studierende

Bei Beginn des Studiums im Sommersemester ist zu beachten, dass die in der nachfolgenden Semestereinteilung mit Stern markierten Lehrveranstaltungen eine positiv absolvierte Studieneingangs- und Orientierungsphase (= Lehrveranstaltungen des ersten Semesters) voraussetzen. Daher ist ein schiefsemestriger Einstieg nur jenen Studierenden anzuraten, die in der Lage sind, sämtliche Lehrveranstaltungen des ersten Semesters bis Beginn des 2.Semesters positiv abzuschließen.

1. Semester (SS)

- 4.0/4.0 VO Algebra und Diskrete Mathematik für Informatik und Wirtschaftsinformatik
- 5.0/2.0 UE Algebra und Diskrete Mathematik für Informatik und Wirtschaftsinformatik
- 3.0/2.0 VU Datenmodellierung
- 3.0/2.0 VU Formale Modellierung
- 2.9/2.0 VU Grundlagen der Programmkonstruktion
- 5.9/4.0 UE Programmierpraxis
- 0.2/1.0 UE Studieneingangsgespräch
- 6.0/4.0 VU Technische Grundlagen der Informatik

2. Semester (WS)

- 2.0/2.0 VO Analysis für Informatik und Wirtschaftsinformatik
- 4.0/2.0 UE Analysis für Informatik und Wirtschaftsinformatik
- * 3.0/2.0 VU Daten- und Informatikrecht
- * 3.0/2.0 VU Funktionale Programmierung
- * 3.0/2.0 VU Gesellschaftswissenschaftliche Grundlagen der Informatik
- * 3.0/2.0 VU Introduction to Security
- * 3.0/2.0 VU Objektorientierte Programmiertechniken
- * 3.0/2.0 VO Statistik und Wahrscheinlichkeitstheorie
- * 3.0/2.0 UE Statistik und Wahrscheinlichkeitstheorie

3. Semester (SS)

- 6.0/4.0 VU Algorithmen und Datenstrukturen 1
- 3.0/2.0 VU Algorithmen und Datenstrukturen 2
- 3.0/2.0 VU Basics of Human Computer Interaction
- 6.0/5.0 VU Einführung in Visual Computing
- 3.0/2.0 VU Gesellschaftliche Spannungsfelder der Informatik
- 3.0/2.0 VU Objektorientierte Modellierung
- * 6.0/4.0 VU Theoretische Informatik und Logik

4. Semester (WS)

- * 2.0/2.0 VO Betriebssysteme
- * 4.0/2.0 UE Betriebssysteme
- * 6.0/4.0 VU Datenbanksysteme
- * 5.0/3.0 VU Einführung in wissensbasierte Systeme
- * 3.0/2.0 VO Verteilte Systeme
- * 3.0/2.0 UE Verteilte Systeme
- * 3.0/2.0 SE Wissenschaftliches Arbeiten

5. Semester (SS)

- * 3.0/2.0 VU Einführung in die Künstliche Intelligenz
- * 3.0/2.0 VO Software Engineering und Projektmanagement
- * 6.0/4.0 PR Software Engineering und Projektmanagement

6. Semester (WS)

- * 3.0/2.0 VU Interface and Interaction Design
- * 10.0/5.0 PR Bachelorarbeit für Informatik und Wirtschaftsinformatik