

Modulhandbuch
Bachelor-Studiengang
Informatik
Prüfungsordnung 20.0

Wedel, den 16. Dezember 2021

Teil I

Modulhandbuch

Kapitel 1.1

Modulhandbuch

Modulverzeichnis nach Modulkürzel

| | |
|--|-----|
| B001 Analysis | 32 |
| B002 Mathematische Konzepte und Diskrete Mathematik | 37 |
| B003 Programmstrukturen 1 | 14 |
| B004 Informationstechnik | 23 |
| B006 Einführung in die Digitaltechnik | 27 |
| B010 Grundlagen der Funktionalen Programmierung | 19 |
| B019 Deskriptive Statistik und Grundlagen der Linearen Algebra | 63 |
| B020 Programmstrukturen 2 | 41 |
| B023 Rechnerstrukturen und Digitaltechnik | 57 |
| B034 Einführung in die Betriebswirtschaft | 103 |
| B036 Programmierpraktikum | 48 |
| B037 Rechnernetze | 89 |
| B040 Algorithmen und Datenstrukturen | 73 |
| B041 Induktive Statistik | 135 |
| B042 Datenschutz und Wirtschaftsprivatrecht | 69 |
| B043 Systemnahe Programmierung | 79 |
| B044 UNIX und Shell-Programmierung | 125 |
| B045 Lineare Algebra | 98 |
| B052 Einführung in Datenbanken | 84 |
| B057 Fortgeschrittene Objektorientierte Programmierung | 108 |
| B058 Software-Design | 116 |
| B059 Web-Anwendungen | 120 |
| B082 Operations Research | 175 |
| B085 Grundlagen der Computergrafik | 180 |
| B087 Systemmodellierung | 148 |
| B093 Softwarequalität | 143 |
| B095 Anwendungen der Künstlichen Intelligenz | 140 |
| B098 Anwendungsentwicklung in ERP-Systemen | 157 |
| B099 Auslandssemester | 196 |
| B101 Echtzeitsysteme | 184 |
| B117 Datenbanktheorie und -implementierung | 113 |
| B118 Soft Skills | 165 |
| B121 Software-Projekt | 170 |
| B122 IT-Sicherheit | 160 |
| B150 Bachelor-Thesis | 207 |
| B159 Betriebspraktikum | 205 |
| B160 Bachelor-Kolloquium | 210 |
| B176 Praxissemester (dual) | 198 |

| | |
|---|-----|
| B179 Wissenschaftliche Ausarbeitung (dual)..... | 202 |
| B209 Applied Data Science and Machine Learning..... | 191 |
| B232 Formale Sprachen | 52 |
| B253 Geschäftsprozesse mit ERP-Systemen | 130 |
| B255 Seminar Informatik | 154 |

Modulverzeichnis nach Modulbezeichnung

| | |
|---|-----|
| Algorithmen und Datenstrukturen | 73 |
| Analysis | 32 |
| Anwendungen der Künstlichen Intelligenz | 140 |
| Anwendungsentwicklung in ERP-Systemen | 157 |
| Applied Data Science and Machine Learning | 191 |
| Auslandssemester | 196 |
| Bachelor-Kolloquium | 210 |
| Bachelor-Thesis | 207 |
| Betriebspraktikum | 205 |
| Datenbanktheorie und -implementierung | 113 |
| Datenschutz und Wirtschaftsprivatrecht | 69 |
| Deskriptive Statistik und Grundlagen der Linearen Algebra | 63 |
| Echtzeitsysteme | 184 |
| Einführung in Datenbanken | 84 |
| Einführung in die Betriebswirtschaft | 103 |
| Einführung in die Digitaltechnik | 27 |
| Formale Sprachen | 52 |
| Fortgeschrittene Objektorientierte Programmierung | 108 |
| Geschäftsprozesse mit ERP-Systemen | 130 |
| Grundlagen der Computergrafik | 180 |
| Grundlagen der Funktionalen Programmierung | 19 |
| Induktive Statistik | 135 |
| Informationstechnik | 23 |
| IT-Sicherheit | 160 |
| Lineare Algebra | 98 |
| Mathematische Konzepte und Diskrete Mathematik | 37 |
| Operations Research | 175 |
| Praxissemester (dual) | 198 |
| Programmierpraktikum | 48 |
| Programmstrukturen 1 | 14 |

| | |
|---|-----|
| Programmstrukturen 2 | 41 |
| Rechnernetze | 89 |
| Rechnerstrukturen und Digitaltechnik | 57 |
| Seminar Informatik | 154 |
| Soft Skills | 165 |
| Software-Design | 116 |
| Software-Projekt | 170 |
| Softwarequalität | 143 |
| Systemmodellierung | 148 |
| Systemnahe Programmierung | 79 |
| UNIX und Shell-Programmierung | 125 |
| Web-Anwendungen | 120 |
| Wissenschaftliche Ausarbeitung (dual) | 202 |

I.1.1 Erläuterungen zu den Modulbeschreibungen

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

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

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

Angaben zum Modul

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

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

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

Angaben zu den Lehrveranstaltungen

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

I.1.2 Programmstrukturen 1

B003 Programmstrukturen 1

| | |
|---------------------------------|--|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B003 |
| Bezeichnung | Programmstrukturen 1 |
| Lehrveranstaltung(en) | B003a Programmstrukturen 1 B003b Übg. Programmstrukturen 1 |
| Verantwortliche(r) | Prof. Dr. Andreas Häuslein |
| Zuordnung zum Curriculum | Computer Games Technology (Bachelor) Data Science & Artificial Intelligence (Bachelor) E-Commerce (Bachelor) IT-Ingenieurwesen (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Smart Technology (Bachelor) Technische Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) |
| Verwendbarkeit | Das Modul ist ein Einführungsmodul in den Themenbereich Programmierung für alle Studiengänge mit Informatikbezug. Die erworbenen Kompetenzen sind insbesondere die Grundlage für das Modul "Programmstrukturen 2", aber auch für die Module "Systemnahe Programmierung" und "UNIX und Shell-Programmierung". |
| Semesterwochenstunden | 10 |
| ECTS | 5.0 |
| Voraussetzungen | Grundkenntnisse der Schulmathematik, Basisfähigkeit zum abstrakten Denken. |
| Dauer | 1 |

Lernziele

Nach Abschluss des Moduls verfügen die Studierenden über Grundkenntnisse hinsichtlich der Entwicklung von Programmen. Studierende mit Vorkenntnissen im Bereich der Programmierung sind in der Lage, diese fachlich fundiert einzuordnen und zu ergänzen.

Nach Abschluss des Moduls sind die unterschiedlichen Vorkenntnisse angeglichen und es ist eine gemeinsame Kompetenzbasis für die weiteren Veranstaltungen im thematischen Umfeld

der Programmierung und Software-Entwicklung gelegt.

Die Studierenden beherrschen sowohl die grundlegenden theoretischen Aspekte der Programmierung als auch die Basiskonzepte von imperativen, prozeduralen Programmiersprachen: Sie kennen alle wesentlichen Anweisungen zur Umsetzung algorithmischer Strukturen ebenso wie die typischen einfachen und strukturierten Datentypen. Dies schließt die Kenntnis einfacher dynamischer Datenstrukturen (dyn. Listen) hinsichtlich ihres Aufbaus und ihrer Verarbeitung mit ein. Die Studierenden kennen die Strukturierungsmöglichkeiten, die durch Prozeduren und Funktionen eröffnet werden und können diese zur Strukturierung ihrer Programme angemessen einsetzen. Die Studierenden können auf Basis dieser Kenntnis die programmiersprachlichen Mittel problemadäquat bei der Formulierung von Programmtexten nutzen.

Sie sind in der Lage, vollständige Programme begrenzter Komplexität eigenständig zu entwickeln und dabei die funktionale Korrektheit der Software sicherzustellen.

Die Studierenden kennen die typischen Funktionen einer Integrierten Entwicklungsumgebung und können diese angemessen zur Software-Entwicklung nutzen.

I.1.2.1 Programmstrukturen 1

| | |
|---------------------------------|---|
| Lehrveranstaltung | Programmstrukturen 1 |
| Dozent(en) | Andreas Häuslein |
| Hörtermin | 1 |
| Häufigkeit | jedes Semester |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 4 |
| ECTS | 3.0 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, E-Learning, Handout, interaktive Entwicklung und Diskussion von Modellen, Online-Aufbereitung, Overheadfolien, Softwaredemonstration, Tafel |

Lernziele

Die Studierenden ...

- kennen die grundlegenden Konzepte imperativer Programmiersprachen und ihre Umsetzung in der Programmiersprache Pascal und können diese benennen.
- kennen die Syntax, Semantik und Pragmatik als wesentliche Aspekte von Programmiersprachen und können diese unterscheiden.
- kennen die wichtigsten Sprachbestandteile der Programmiersprache Pascal und beschreiben diese.
- setzen die Konzepte und Sprachbestandteile angemessen zur Lösung von Problemstellungen begrenzter Komplexität ein und bauen vollständige Programme für diese Problemstellungen auf.
- kennen die wesentlichen statischen Datenstrukturen imperativer Programmiersprachen, wählen bei der Programmierung zwischen diesen in Abhängigkeit von der Aufgabenstellung sicher aus und setzen sie angemessen zur Realisierung der Programmfunktionalität ein.
- kennen die Realisierung einfacher dynamischer Datenstrukturen und können diese zur Realisierung von Algorithmen nutzen.
- kennen wesentliche Qualitätskriterien für Software und können diese bei der Software-Entwicklung berücksichtigen.
- führen eine Fehlersuche und -beseitigung (Debugging) bei ihren Programmtexten durch.

Inhalt

- Grundkonzepte der Datenverarbeitung
- Entwurf und Darstellung von Algorithmen
- Allgemeine Aspekte von Programmiersprachen

- Daten in Programmen
 - Grundlegende Datentypen
 - Variablen, Zuweisungen, Konstanten
- Grundsätzlicher Aufbau von Programmen
- Operatoren und Ausdrücke
- Einfache und strukturierte Anweisungen
- Statische strukturierte Datentypen und ihre Nutzung
 - Strings
 - Arrays
 - Records
 - Sets
- Zeigertypen
 - Besonderheiten und Probleme bei der Nutzung von Zeigertypen
 - Aufbau dynamischer Datenstrukturen mit Hilfe von Zeigertypen
- Strukturierung von Programmen
 - Prozeduren und Funktionen
 - Units

Literatur

- OTTMANN, Thomas; WIDMAYER, Peter:
Programmierung mit PASCAL: Eine Einführung für Programmieranfänger, 9. Aufl., Springer Vieweg, 2018
- Collingbourne, Huw:
The Little Book Of Delphi Programming: Learn To Program with Object Pascal, Dark Neon, 2020
- CANTU, Marco:
Object Pascal Handbook, CreateSpace Independent Publishing Platform, 2015
- GUMM, Heinz-Peter; SOMMER, Manfred:
Einführung in die Informatik.
11. Aufl. München: Oldenbourg Wissenschaftsverlag, 2013.
- MATTHÄUS, Wolf-Gert:
Grundkurs Programmieren mit Delphi: Systematisch programmieren lernen für Einsteiger, 5. Aufl., Springer Vieweg, 2016
- WIRTH, Niklaus:
Algorithmen und Datenstrukturen: Pascal-Version. 5. Aufl., Teubner-Verlag, 2013

I.1.2.2 Übg. Programmstrukturen 1

| | |
|---------------------------------|--|
| Lehrveranstaltung | Übg. Programmstrukturen 1 |
| Dozent(en) | Lars Neumann |
| Hörtermin | 1 |
| Häufigkeit | jedes Semester |
| Lehrform | Übung/Praktikum/Planspiel |
| Semesterwochenstunden | 6 |
| ECTS | 2.0 |
| Prüfungsform | Abnahme |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, Softwaredemonstration, studentische Arbeit am Rechner, Tafel, Tutorien |

Lernziele

Die Studierenden ...

- festigen und vertiefen ihr Wissen zu den in der zugehörigen Vorlesung „Programmstrukturen 1“ vorgestellten Konzepten
- beherrschen die Arbeit mit einer modernen Entwicklungsumgebung (Embarcadero Delphi 10.3)
- lernen Grundlagen des Debugging und der Versionsverwaltung kennen
- erweitern ihre Teamfähigkeit durch die eigenständige praktische Anwendung des erlernten Wissens in Zweiergruppen

Inhalt

Ausgehend von den Grundlagen der Programmierung wie Datentypen, Verzweigungen und Iterationen werden in der Übung Programmstrukturen 1 in den einzelnen Aufgaben Ein- und Ausgabe, Operatoren, Bedingungen, Schleifen, Strings (sowohl über Stringfunktionen als auch über indizierten Zugriff), Arrays, Records, Mengen, Prozeduren und Funktionen, Zeiger und Listen sowie Dateien und Exceptions behandelt.

Die Inhalte höherer Aufgaben schließen dabei in der Regel die Inhalte der vorherigen mit ein.

Literatur

Skript:

- Vorlesungsskript unter <https://stud.fh-wedel.de/handout/Haeuslein/Programmstrukturen%201/>
- Weiteres Material unter <http://www.fh-wedel.de/mitarbeiter/ne/uebung-programmstrukturen-1/>

I.1.3 Grundlagen der Funktionalen Programmierung

B010 Grundlagen der Funktionalen Programmierung

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B010 |
| Bezeichnung | Grundlagen der Funktionalen Programmierung |
| Lehrveranstaltung(en) | B010a Grundlagen der Funktionalen Programmierung B010b Übg. Grundlagen der Funktionalen Programmierung |
| Verantwortliche(r) | Priv.-Doz. Dr. Frank Huch |
| Zuordnung zum Curriculum | Informatik (Bachelor) |
| Verwendbarkeit | Das Modul ist eine sinnvolle Grundlage für alle Module über Programmiersprachen und Softwaretechnik. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Außer elementarer Schulmathematik sind keine Vorkenntnisse für dieses Modul erforderlich. |
| Dauer | 1 |

Lernziele

Nach Abschluss des Moduls kennen die Studierenden das Paradigma des funktionalen Programmierens und können damit praktisch arbeiten.

Sie kennen die Einfachheit und Eleganz der Konstruktion von Algorithmen durch das Weglassen von Programmvariablen, Zuweisungen und Schleifen. Sie können diese durch praktische Übungen auch konkret nachvollziehen. Sie beherrschen die Konstruktion rekursiver Algorithmen und kennen die Unterschiede zwischen Endrekursion, linearer Rekursion und allgemeineren Rekursionsformen.

Neben Korrektheitsargumentationen verfügen sie ferner über ein erstes Grundverständnis über das Führen solcher Argumentationen, wie Terminierung und Ableitung von Gesetzen und Programmeigenschaften. Sie kennen die Vorteile beim Entwickeln in der Programmiersprache Haskell bezüglich einer streng getypten und auf mathematischen Konzepten basierenden Programmiersprache.

Sie verfügen über die Fähigkeiten zur Abstraktion und zur Modellbildung. Ferner verfügen sie über eine Basis für ein sorgfältiges, systematisches und methodisches Vorgehen beim Programmieren im Kleinen.

I.1.3.1 Grundlagen der Funktionalen Programmierung

| | |
|---------------------------------|--|
| Lehrveranstaltung | Grundlagen der Funktionalen Programmierung |
| Dozent(en) | Uwe Schmidt |
| Hörtermin | 1 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 2 |
| ECTS | 2.0 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Softwaredemonstration |

Lernziele

Die Studierenden ...

- kennen die Programmierparadigmen der funktionalen Programmierung und der Algorithmenkonstruktion ohne Programmvariablen, Zuweisungen und Schleifen.
- besitzen das Verständnis und den Umgang mit Rekursion beim Entwickeln von Algorithmen.
- kennen die Vorteile einer streng getypten Programmiersprache bei der Programmentwicklung und beim Erlernen einer neuen Programmiersprache.
- erwerben das erste Grundverständnis über das Führen von Korrektheitsargumentationen wie Terminierung und Ableitung von Gesetzen und Programmeigenschaften.
- sind geschult bezüglich der Fähigkeiten zur Abstraktion und zur Modellbildung.
- trainieren das sorgfältige, methodische und systematische Programmieren im Kleinen.

Inhalt

- Einleitung
- Erste Schritte
- Typen und Klassen
- Funktionsdefinitionen
- Rekursive Funktionen
- Funktionen höherer Ordnung
- Korrektheitsargumentationen
- Selbstdefinierte Datenstrukturen und Bäume

Literatur

- Uwe Schmidt:
Funktionale Programmierung,

Vorlesungsunterlagen im Web: <http://www.fh-wedel.de/~si/vorlesungen/fp/fp.html>

- Graham Hutton:
Programming in Haskell
Cambridge University Press, 2007, ISBN-13: 9780511292187

I.1.3.2 Übg. Grundlagen der Funktionalen Programmierung

| | |
|---------------------------------|---|
| Lehrveranstaltung | Übg. Grundlagen der Funktionalen Programmierung |
| Dozent(en) | Malte Heins |
| Hörtermin | 1 |
| Häufigkeit | jährlich |
| Lehrform | Übung/Praktikum/Planspiel |
| Semesterwochenstunden | 2 |
| ECTS | 3.0 |
| Prüfungsform | Abnahme |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, studentische Arbeit am Rechner, Tafel, Tutorien |

Lernziele

Die Studierenden ...

- wenden praktisch die Inhalte aus der Vorlesung an.
- beherrschen die grundlegenden Sprachelemente der Programmiersprache Haskell und die grundlegenden Konzepte der funktionalen Programmierung.
- erlangen die Fähigkeit zur selbständigen Erstellung kleiner, aber nicht trivialer, Algorithmen einschließlich systematischer Tests dieser Programmteile.

Inhalt

Bearbeitung von Übungsaufgaben parallel zum Stoff der Vorlesung in Zweiergruppen mit Abnahme und Diskussion der Lösungen. Zusätzlich werden praxisrelevante Aspekte der Anwendungsentwicklung mit der Programmiersprache Haskell behandelt, die nicht Bestandteil der Vorlesung sind wie z.B. Modultests in Haskell.

Literatur

- Unterlagen zur Übung im Web
- siehe auch Vorlesung Grundlagen der Funktionalen Programmierung

I.1.4 Informationstechnik

B004 Informationstechnik

| | |
|---------------------------------|--|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B004 |
| Bezeichnung | Informationstechnik |
| Lehrveranstaltung(en) | B004a Informationstechnik |
| Verantwortliche(r) | PD Dr. Dennis Säring |
| Zuordnung zum Curriculum | Computer Games Technology (Bachelor) Data Science & Artificial Intelligence (Bachelor) IT-Ingenieurwesen (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Smart Technology (Bachelor) Technische Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) Wirtschaftsingenieurwesen (Bachelor) |
| Verwendbarkeit | Das Modul "Informationstechnik" ist ein Einführungsmodul und soll ein breites Grundverständnis für die Funktionsweise von Rechnern vermitteln. Die erworbenen Kompetenzen stellen damit die Grundlagen für zum Beispiel die Module "Rechnerstrukturen und Digitaltechnik", "Systemsoftware" und "Großintegrierte Systeme" dar. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Keine |
| Dauer | 1 |

Lernziele

Grundlegendes Verständnis für die Funktionsweise von Rechnern, sowohl aus Sicht der technischen Vorgänge und technischen Funktionselemente als auch aus informationstheoretischer Sicht.

Kenntnisse der rechnerinternen Abläufe auf allen technischen Beschreibungsebenen: vom Transistor, über Logikgatter und Schaltnetzen, hin zu Prozessorstrukturen, der Maschinenbefehlsebene und der Hochsprachenbefehlsebene.

Verständnis des quantitativen Informationsbegriffs und unterschiedlichen Kodierungsmöglichkeiten von Informationen, sowohl verlustfrei als auch verlustbehaftet.

Wissen um alternative Informationsverarbeitende Ansätze, die sich stark von der von-Neumann-Architektur unterscheiden.

I.1.4.1 Informationstechnik

| | |
|---------------------------------|---------------------|
| Lehrveranstaltung | Informationstechnik |
| Dozent(en) | Dennis Säring |
| Hörtermin | 1 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | E-Learning |

Lernziele

Die Studierenden ...

- besitzen grundlegende Kompetenzen zum Verständnis der Funktionalität von Rechnern in Bezug auf ihre informationstheoretischen Grundlagen und deren praktische Implementierung
- können Vorgänge der Informationsverarbeitung auf der Maschinenebene theoretisch sowie praktisch umsetzen
- sind in der Lage die Umsetzung von Befehlen höherer Sprachebenen in Maschinenbefehle und in deren rechnerinternen Interpretation nachzuvollziehen
- kennen die Ansätze aktueller Rechnerstrukturen und Kommunikationsschnittstellen mit der Peripherie
- sind vertraut mit informationstheoretischen Ansätzen und unterschiedlichen Kodierungsverfahren.

Inhalt

- Grundlagen der Halbleitertechnik
- Logikgatter und Schaltnetze
- Zahlendarstellung und Berechnung
- FlipFlop und weitere Speicherstrukturen
- Moderne Rechnerarchitekturen
- Programmcode zu Assembler
- Computerperipherie
- Informationstheorie und Kodierung

Literatur

- Gumm, Hans-Peter; Sommer, Manfred: Einführung in die Informatik, Oldenbourg, 8. Auflage 2009.

- Müller, Käser, et., al. :Technische Informatik 1, vdf-Hochschulverlag Zürich, 2003
- Schiffmann, Schmitz: Technische Informatik 2, Grundlagen der Computertechnik, Springer-Verlag 1998
- Martin: Einführung in die Rechnerarchitektur, Fachbuchverlag Leipzig, 2003

I.1.5 Einführung in die Digitaltechnik

B006 Einführung in die Digitaltechnik

| | |
|---------------------------------|--|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B006 |
| Bezeichnung | Einführung in die Digitaltechnik |
| Lehrveranstaltung(en) | B006a Einführung in die Digitaltechnik B006b Prakt. Digitaltechnik |
| Verantwortliche(r) | Prof. Dr. Sergei Sawitzki |
| Zuordnung zum Curriculum | IT-Ingenieurwesen (Bachelor) Informatik (Bachelor) Smart Technology (Bachelor) Technische Informatik (Bachelor) |
| Verwendbarkeit | Das Modul "Einführung in Digitaltechnik" ist ein Einführungsmodul. Die erworbenen Kompetenzen stellen Grundlagen für zum Beispiel die Module "Rechnerstrukturen und Digitaltechnik", "Diskrete Systeme" und "Systementwurf mit VHDL" dar. Grundsätzlich kann das Modul sinnvoll mit den Modulen kombiniert werden, die ein Rechnersystem auf höheren Abstraktionsebenen (über dem Gatterniveau) behandeln. Das Modul ist fachübergreifend für alle Studiengänge relevant, die eine grundlegende Hardware-Kompetenz voraussetzen und hat somit eine direkte Verbindung zu den Studiengangszielen des Studiengangs "Bachelor Technische Informatik". |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Es werden mathematische Grundlagenkenntnisse entsprechend der Abitur-Stufe (überwiegend diskrete Mathematik) sowie das Verständnis einfacher technischer naturwissenschaftlicher Zusammenhänge vorausgesetzt. |
| Dauer | 1 |
| Lernziele | Das Ziel des Moduls besteht in Vermittlung der allgemeinen Konzepte und Prinzipien des Aufbaus, der Funktionsweise sowie der Analyse und des Entwurfs digitaler Systeme. Angefangen bei Grundbegriffen (analoge und digitale Signale und Systeme, Schaltvariablen, Schaltalgebra usw.) lernen die Studierende Schaltnetze als technische Umsetzung von Schaltfunktionen |

kennen. Als inhaltliche Voraussetzung für das Modul "Rechnerstrukturen und Digitaltechnik" werden schließlich die Speicherelemente diskutiert. Somit sind Studierende nach dem Abschluss des Moduls auf die Betrachtung komplexer, zustandsbehafteter Systeme vorbereitet, mit der das Modul "Rechnerstrukturen und Digitaltechnik" beginnt.

I.1.5.1 Einführung in die Digitaltechnik

| | |
|---------------------------------|------------------------------------|
| Lehrveranstaltung | Einführung in die Digitaltechnik |
| Dozent(en) | Sergei Sawitzki |
| Hörtermin | 1 |
| Häufigkeit | jedes Semester |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 2 |
| ECTS | 3.0 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, Handout, Tafel |

Lernziele

Die Studierenden ...

- gewinnen eine Einsicht in die mathematischen Grundlagen der digitalen Schaltungstechnik
- erlernen die Methoden zur Analyse, Darstellung und Vereinfachung von Schaltfunktionen
- begreifen ein Schaltnetz als technische Umsetzung einer Schaltfunktion
- lernen die wichtigsten Grundelemente digitaler Systeme kennen
- erwerben die Fähigkeit, einfache digitale Systeme zu begreifen, zu spezifizieren, zu entwerfen und zu optimieren

Inhalt

- Einleitung: Digitale Systeme
- Mathematische Grundlagen
 - Entstehungsgeschichte
 - Aussagenlogik und Boolesche Algebra
 - Schaltalgebra, Schaltfunktionen und Schaltfunktionssysteme
 - Operatorensysteme
 - Normalformen und Dualitätsprinzip
- Schaltnetze
 - Darstellung
 - Vereinfachung (KV-Diagramme, QMCV, BDDs)
 - Analyse (Funktion, Komplexität, Zeitverhalten)
 - Synthese und Realisierung
 - Beispiele

- Speicherelemente

Literatur

- Hoffmann, Dirk: Grundlagen der technischen Informatik, Carl Hanser Verlag 2007
- Schiffmann, Wolram; Schmitz, Robert: Technische Informatik, in 3 Bänden. 3. Auflage Springer Verlag, 1996
- Beuth, Klaus: Elektronik 4. Digitaltechnik, 13. Auflage Vogel Verlag und Druck 2003

I.1.5.2 Prakt. Digitaltechnik

| | |
|---------------------------------|-------------------------------|
| Lehrveranstaltung | Prakt. Digitaltechnik |
| Dozent(en) | Dieter Opitz |
| Hörtermin | 1 |
| Häufigkeit | jedes Semester |
| Lehrform | Übung/Praktikum/Planspiel |
| Semesterwochenstunden | 2 |
| ECTS | 2.0 |
| Prüfungsform | Praktikumsbericht / Protokoll |
| Sprache | deutsch |
| Lehr- und Medienform(en) | |

Lernziele

Fähigkeit zur Übertragung des theoretischen Wissens in eine erlebbare, reale technische Umgebung.

Inhalt

- Schaltnetzentwurf Schaltnetz wird aus einer gegebenen Spezifikation formal entworfen. Der Entwurf wird auf einem IC-Trainer realisiert. Die Schaltung wird auf Funktion und Einhaltung der Spezifikation überprüft. Die Ergebnisse werden dokumentiert.
- Schaltwerkentwurf Schaltwerk (z. B. ein Zähler) wird aus einer gegebenen Spezifikation formal entworfen. Der Entwurf wird auf einem IC-Trainer realisiert. Die Schaltung wird auf Funktion und Einhaltung der Spezifikation überprüft. Die Ergebnisse werden dokumentiert.

Literatur

Vorlesungsskript

I.1.6 Analysis

B001 Analysis

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B001 |
| Bezeichnung | Analysis |
| Lehrveranstaltung(en) | B001a Analysis B001b Übg. Analysis |
| Verantwortliche(r) | Dr. Hendrik Glowatzki |
| Zuordnung zum Curriculum | Betriebswirtschaftslehre (Bachelor) Computer Games Technology (Bachelor) Data Science & Artificial Intelligence (Bachelor) E-Commerce (Bachelor) IT-Ingenieurwesen (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Smart Technology (Bachelor) Technische Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) Wirtschaftsingenieurwesen (Bachelor) |
| Verwendbarkeit | Das Modul ist sinnvoll mit anderen Modulen der Mathematik zu kombinieren und zur Bildung mathematischer Grundlagenkompetenzen in allen naturwissenschaftlichen, ingenieurtechnischen und wirtschaftswissenschaftlichen Studiengängen verwendbar. Es stellt Querbezüge zur Finanzmathematik, Linearen Algebra, Statistik, Physik und Betriebswirtschaftslehre her. |
| Semesterwochenstunden | 6 |
| ECTS | 5.0 |
| Voraussetzungen | Voraussetzung zur Teilnahme am Modul sind schulische Grundlagen der Mathematik. Insbesondere gehören hierzu die grundlegenden Begriffe über Mengen, das Rechnen mit reellen Zahlen, Gleichungen mit einer Unbekannten, Basiswissen zur elementaren Geometrie sowie zu Funktionen und Kurven. |
| Dauer | 1 |

Lernziele

Nachdem erfolgreichen Abschluss des Moduls besitzen die Studierenden Rechenfertigkeiten, anschauliche Vorstellungen und theoretisches Verständnis von Funktionen. Sie können dieses auf Funktionen einer reellen Veränderlichen anwenden, Problemstellungen und Lösungswege klassifizieren und bewerten sowie Problemlösungen prüfen und beurteilen. Sie beherrschen die Grundbegriffe der Analysis einer Veränderlichen, können dieses auf Funktionen mehrere Veränderlicher übertragen und als Fundament für die weiteren fachwissenschaftlichen Studien nutzen. Sie verfügen über formalisierte mathematische Denk- und Arbeitsweisen und sind befähigt mathematische Kausalzusammenhänge aufzustellen und sich in neue formale Systeme einzuarbeiten. Die Studierenden erkennen die Querbezüge der Analysis zu anderen mathematischen und fachspezifischen Fächern

Durch die Übungen erarbeiten sie sich einen sicheren, präzisen und selbständigen Umgang mit den in den Vorlesungen behandelten Begriffen, Aussagen und Methoden. Praxisorientierte Problemstellungen können sie in mathematische Beziehungen und Modelle umsetzen und anhand dieser Modelle bearbeiten. Sie können die Praxisrelevanz der Analysis für verschiedene Fachgebieten bewerten und die Analysis auf Problemstellungen aus Informatik, Technik und Ökonomie anwenden.

I.1.6.1 Analysis

| | |
|---------------------------------|-------------------|
| Lehrveranstaltung | Analysis |
| Dozent(en) | Hendrik Glowatzki |
| Hörtermin | 1 |
| Häufigkeit | jedes Semester |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 4 |
| ECTS | 3.0 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Handout, Tutorien |

Lernziele

Die Studierenden ...

- kennen und verstehen die grundlegenden Begriffe, Aussagen und Methoden der Analysis,
- können mathematische Regeln korrekt anwenden,
- verstehen Beweistechniken,
- erkennen die fundamentale Bedeutung des Grenzwertbegriffes für die Analysis,
- beherrschen die Methoden des Differenzierens und Integrierens,
- können die eindimensionale Differentialrechnung bei praxisorientierten Fragestellungen flexibel in unterschiedlichen Fachgebieten einsetzen und dabei beurteilen, welche analytischen Hilfsmittel für welche Problemstellungen zielführend sind,
- erkennen die Anwendbarkeit und den Nutzen der Analysis für unterschiedliche Fachgebiete und deren spezifischen Problemstellungen,
- können praxisorientierte Problemstellungen in mathematische Beziehungen bzw. Modelle umzusetzen und anhand analytischer Modelle weiter bearbeiten
- können neue, unklare und ungewöhnliche Aufgabenstellungen als solche erkennen und zur Bearbeitung weiterführende Hilfestellung in Anspruch nehmen,
- verfügen über gesteigerte Kompetenzen sich Fähigkeit durch Selbststudium anzueignen und sich in neue formale Systeme einzuarbeiten

Inhalt

- Zahlentypen
- Folgen
 - Bildungsgesetze
 - Grenzwerte
- Funktionen, Relationen
 - Funktionstypen

- Umkehrfunktion
- Differentialrechnung
 - Differentiationsregeln
 - Anwendungen der Differentialrechnung (Kurvendiskussionen und Extremwerte)
- Integralrechnung
 - Integrationsmethoden
 - Anwendungen der Integralrechnung (Bestimmte Integrale)
- Funktionen mit zwei Variablen
 - Partielle Differentiation
 - Extremwertaufgaben mit Nebenbedingungen

Literatur

- BÖHME, Gert:
Analysis 1.
6. Aufl. Berlin: Springer-Verlag, 1990
- FETZER, Albert; FRÄNKEL, Heiner:
Mathematik 1.
10. bearbeitete Aufl. Berlin: Springer-Verlag, 2008
- FETZER, Albert; FRÄNKEL, Heiner:
Mathematik 2.
6. korrigierte Aufl.. Berlin: Springer-Verlag, 2009
- HENZE, Norbert; Last, Günter:
Mathematik für Wirtschaftsingenieure 1.
2. Aufl. Wiesbaden: Vieweg + Teubner Verlag, 2005
- KUSCH, Lothar:
Mathematik. Aufgabensammlung mit Lösungen. Bd. 3
9. Aufl. Berlin: Cornelsen Verlag, 1995
- OHSE, Dietrich: Mathematik für Wirtschaftswissenschaftler 1. Analysis.
6. Aufl. München: Verlag Vahlen, 2004
- PAPULA, Lothar:
Mathematik für Ingenieure und Naturwissenschaftler 1: Ein Lehr- und Arbeitsbuch für
das Grundstudium.
12. überarbeitete und erweiterte Aufl. Wiesbaden: Vieweg + Teubner Verlag, 2009
- PREUSS, Wolfgang; WENISCH, Günter:
Lehr- und Übungsbuch Mathematik 1: Grundlagen - Funktionen - Trigonometrie.
2. neu bearbeitete Aufl. München: Carl Hanser Verlag, 2003
- PREUSS, Wolfgang; WENISCH, Günter:
Lehr- und Übungsbuch Mathematik 2: Analysis.
3. Aufl. München: Carl Hanser Verlag, 2003

I.1.6.2 Übg. Analysis

| | |
|---------------------------------|---------------------------|
| Lehrveranstaltung | Übg. Analysis |
| Dozent(en) | Fikret Koyuncu |
| Hörtermin | 1 |
| Häufigkeit | jedes Semester |
| Lehrform | Übung/Praktikum/Planspiel |
| Semesterwochenstunden | 2 |
| ECTS | 2.0 |
| Prüfungsform | Teilnahme |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Handout, Tafel |

Lernziele

Die Studierenden können ...

- praktische Problemstellungen mathematisch formulieren
- beurteilen, welche analytischen Hilfsmittel zielführend sind
- neue, unklare und ungewöhnliche Aufgabenstellungen als solche erkennen und mit weiterführender Hilfestellung bearbeiten
- Lösungsansätze präsentieren und begründen

Inhalt

- Bearbeitung von Übungsaufgaben aus dem Themenspektrum der zugehörigen Lehrveranstaltung
- Vorstellung und Diskussion möglicher Lösungswege
- Heranführung an mathematische Softwaretools

Literatur

PAPULA, Lothar: Mathematik für Ingenieure und Naturwissenschaftler: Klausur- und Übungsaufgaben 4. überarbeitete und erweiterte Aufl. Wiesbaden: Vieweg + Teubner Verlag, 2010

I.1.7 Mathematische Konzepte und Diskrete Mathematik

B002 Mathematische Konzepte und Diskrete Mathematik

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B002 |
| Bezeichnung | Mathematische Konzepte und Diskrete Mathematik |
| Lehrveranstaltung(en) | B002a Diskrete Mathematik |
| Verantwortliche(r) | Prof. Dr. Sebastian Iwanowski |
| Zuordnung zum Curriculum | Computer Games Technology (Bachelor) Data Science & Artificial Intelligence (Bachelor) E-Commerce (Bachelor) IT-Ingenieurwesen (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Smart Technology (Bachelor) Technische Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) |
| Verwendbarkeit | Das Modul ist ein Einführungsmodul. Es liefert die Konzepte für ein tieferes Verständnis der anderen Mathematikmodule wie "Analysis" und "Lineare Algebra". Die vermittelten Konzepte und Inhalte werden gebraucht in den Modulen "Informationstechnik", "Einführung in Digitaltechnik", "Programmstrukturen 1 und 2", "Formale Sprachen", "Algorithmen und Datenstrukturen", "Einführung in Datenbanken" und "Anwendungen der Künstlichen Intelligenz". Außerdem werden die in diesem Modul vermittelten Kenntnisse in allen Mastervorlesungen der IT-orientierten Studiengänge vorausgesetzt. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Die Studierenden müssen auf dem Kenntnisstand der Schulmathematik der 9. Klasse (Gymnasium) sein. Sie sollten insbesondere mit den Mengen der natürlichen, ganzen, rationalen und reellen Zahlen sowie mit den dafür geltenden Rechengesetzen vertraut sein. Außerdem wird ein gutes logisches Denkvermögen vorausgesetzt. |

Dauer

1

Lernziele

Nach Abschluss de Moduls verstehen und beherrschen die Studierenden allgemeine formalisierte mathematische Denk- und Arbeitsweisen. Sie kennen grundlegende Beweistechniken und verstehen die Notwendigkeit mathematischen Beweisens. Ferner verfügen sie über die Fähigkeit, Kausalzusammenhänge nachzuvollziehen und zu erarbeiten. Sie können mathematische Regeln korrekt anwenden. Die Studierenden sind in der Lage, die Verwendbarkeit der vermittelten mathematischen Hilfsmittel auf praktische Problemstellungen kompetent zu beurteilen. Sie können praxisorientierte Problemstellungen in mathematische Beziehungen bzw. Modelle umsetzen und anhand dieser Modelle bearbeiten und lösen. Ferner können sie sich in neue formale Systeme einarbeiten und dessen Regelwerke richtig anwenden. Schließlich besitzen sie die Fähigkeit, neue, unklare und ungewöhnliche Aufgabenstellungen als solche zu erkennen und zu ihrer Bearbeitung weiterführende Hilfestellung in Anspruch zu nehmen. Im Speziellen beherrschen sie die wesentlichen Konzepte der Diskreten Mathematik und können diese auf anwendungsbezogene Problemstellungen in den Gebieten der Informatik, Technik und Wirtschaft anwenden.

I.1.7.1 Diskrete Mathematik

| | |
|---------------------------------|--|
| Lehrveranstaltung | Diskrete Mathematik |
| Dozent(en) | Sebastian Iwanowski |
| Hörtermin | 1 |
| Häufigkeit | jedes Semester |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, Overheadfolien, Softwaredemonstration, Tafel, Tutorien |

Lernziele

Nach Abschluss der Veranstaltung besitzen die Studierenden folgende Kompetenzen:

- Beherrschen der grundlegenden mathematischen Begriffe und Konzepte (Definition, Satz, Beweis) und Fähigkeit zur Unterscheidung derselben.
- Beherrschen der Grundlagen und der Formalisierung logischen Denkens.
- Verständnis elementarer Logik und Mengenlehre und des inneren Zusammenhangs dieser Gebiete.
- Darauf aufbauendes Verständnis von Relationen und Funktionen.
- Fähigkeit, elementare Beweisprinzipien wie vollständige Induktion in verschiedenen Kontexten anzuwenden.
- Beherrschen der grundlegenden Sätze der elementaren Zahlentheorie, Gruppen- und Körpertheorie, Kombinatorik und Graphentheorie und selbständige Anwendung an Beispielen.

Inhalt

- Logik
 - Einführung
 - Aussagenlogik
 - Prädikatenlogik
- Mengenlehre
 - Grundlegende Begriffe und Konzepte
 - Relationen
 - Funktionen
 - Boolesche Algebren
- Beweisführung

- Strukturen der mathematischen Beweisführung
- Vollständige Induktion
- Beweisstrategien
- Zahlentheorie
 - Teilbarkeit
 - Teilen mit Rest
 - Primzahlen
 - Modulare Arithmetik
- Algebraische Strukturen
 - Gruppen
 - Körper
- Kombinatorik
 - Zählformeln für Mengen
 - Permutationen
- Graphentheorie
 - Terminologie und Repräsentation
 - Wege in Graphen
 - Bäume
 - Planare Graphen
 - Färbungen

Literatur

- Sebastian Iwanowski / Rainer Lang:
Diskrete Mathematik mit Grundlagen, Springer 2014, ISBN 978-3-658-07130-1 (Print),
978-3-658-07131-8 (Online)
- Albrecht Beutelspacher / Marc-Alexander Zschiegner:
Diskrete Mathematik für Einsteiger.
Vieweg 2004 (2. Auflage), ISBN 3-528-16989-3
- Norman L. Biggs:
Discrete Mathematics.
Oxford University Press 2002, ISBN 0-19-850717-8
- Neville Dean: Diskrete Mathematik.
Pearson Studium, Reihe "im Klartext" 2003, ISBN 3-8273-7069-8
- Christoph Meinel / Martin Mundhenk:
Mathematische Grundlagen der Informatik.
Teubner 2002 (2. Auflage), ISBN 3-519-12949-3

I.1.8 Programmstrukturen 2

B020 Programmstrukturen 2

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B020 |
| Bezeichnung | Programmstrukturen 2 |
| Lehrveranstaltung(en) | B020a Programmstrukturen 2 B020b Übg. Programmstrukturen 2 |
| Verantwortliche(r) | Prof. Dr. Andreas Häuslein |
| Zuordnung zum Curriculum | Computer Games Technology (Bachelor) Data Science & Artificial Intelligence (Bachelor) E-Commerce (Bachelor) IT-Ingenieurwesen (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Smart Technology (Bachelor) Technische Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) |
| Verwendbarkeit | Das Modul basiert auf den im Modul "Programmstrukturen 1" erworbenen Kompetenzen. Es schafft die Grundlagen für Module der fortgeschrittenen Programmierung in Informatik-Studiengängen, zum Beispiel die Module "Algorithmen und Datenstrukturen", "Fortgeschrittene Objekt-orientierte Programmierung" und "Web-Anwendungen". |
| Semesterwochenstunden | 6 |
| ECTS | 5.0 |
| Voraussetzungen | Kenntnis der Grundkonzepte imperativer Programmiersprachen hinsichtlich der wesentlichen statischen und einfachen dynamischen Datenstrukturen sowie der Anweisungen zur Umsetzung der algorithmischen Grundstrukturen, Fähigkeit zur Erstellung von vollständigen Programmen begrenzter Komplexität. Die erfolgreiche Teilnahme an der Übung "B003b Übg. Programmstrukturen 1" ist Voraussetzung, um an der Übung "B020b Übg. Programmstrukturen 2" teilzunehmen. |
| Dauer | 1 |

Lernziele

Nach Abschluss des Moduls kennen die Studierenden die Basiskonzepte der Objektorientierten Programmierung und können diese Kenntnisse zur Erstellung von objektorientierter Software begrenzten Umfangs einsetzen. Die Studierenden wissen, wie die Programmiersprache Java grundsätzlich aufgebaut ist, sie kennen die grundlegenden Sprachelemente der Programmiersprache Java und können diese sicher zur Realisierung entsprechender algorithmischer Strukturen nutzen.

Die Studierenden kennen die zentralen Konzepte der Objektorientierten Programmierung (z.B. Vererbung und Dynamische Bindung) und können sie zur Realisierung objektorientierter Software angemessen einsetzen. Die Studierenden können Bezüge zwischen der imperativ prozeduralen Sprache Pascal und der objektorientierten Programmiersprache Java herstellen. Sie sind damit in der Lage, wesentliche allgemeine Konzepte von Programmiersprachen zu erkennen und einzuordnen.

Die Studierenden besitzen die Fähigkeit, eine moderne Entwicklungsumgebung zur Software-Erstellung zu nutzen. Sie können mit den Mitteln der objektorientierten Sprache Java einfache rekursive Datenstrukturen (Listen) aufbauen, kennen grundlegende Algorithmen für diese Datenstrukturen und können Variationen dieser Algorithmen eigenständig entwickeln.

Die Studierenden sind in der Lage die Modularisierungskonzepte der Sprache Java, soweit sie Gegenstand der Vorlesung sind, zu einer problemadäquaten Strukturierung eines Programms mittleren Umfangs und begrenzter Funktionalität einzusetzen.

Sie kennen bezogen auf die Gestaltung einer grafischen Benutzungsoberfläche die wesentlichen Regeln und Richtlinien und sind in der Lage diese für die Gestaltung konkreter Oberflächen einzusetzen. Sie besitzen die Kenntnis hinsichtlich einer konkreten technischen Umsetzung von grafischen Oberflächen und können diese zur Implementation solcher Oberflächen nutzen.

Sie verfügen über Basiskenntnisse hinsichtlich der Qualitätssicherung von Software in Form einfacher Testverfahren und können diese einsetzen, um die funktionale Korrektheit und ein ausreichendes Maß an Zuverlässigkeit der Software zu gewährleisten.

I.1.8.1 Programmstrukturen 2

| | |
|---------------------------------|--|
| Lehrveranstaltung | Programmstrukturen 2 |
| Dozent(en) | Andreas Häuslein |
| Hörtermin | 2 |
| Häufigkeit | jedes Semester |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 4 |
| ECTS | 3.0 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, E-Learning, Handout, Online-Aufbereitung, Softwaredemonstration, Tafel |

Lernziele

Die Studierenden ...

- identifizieren die Basiskonzepte der Objektorientierten Programmierung und stellen diese den Konzepten der prozeduralen Programmierung gegenüber.
- entwickeln Software auf der Grundlage der Kernkonzepte der Objektorientierten Programmierung.
- stellen die grundlegenden Sprachelemente (Datentypen, Anweisungen, Realisierung von objektorientierten Konzepten) von Java zusammen und wählen daraus aus, um Java-Programme mittlerer Komplexität zu entwickeln.
- vergleichen die Programmiersprachen Pascal und Java und stellen ihre Gemeinsamkeiten und Unterschiede heraus.
- setzen eine moderne Entwicklungsumgebung zur Unterstützung der Softwareentwicklung ein und stellen die damit verbundenen Funktionalitäten und Vorgehensweisen dar.
- entwerfen einfache dynamische Datenstrukturen im Kontext einer objektorientierten Programmiersprache.
- erläutern grundlegende Algorithmen, die auf den vermittelten Datenstrukturen arbeiten.
- entwerfen für Programme mittlerer Komplexität durch Einsatz geeigneter Elemente der Programmiersprache Java eine angemessene Modularisierung und legen entsprechende Schnittstellen zwischen den Modulen fest.
- benennen die Grundregeln der benutzungsgerechten Gestaltung von Programmen und nutzen diese, um Benutzungsoberflächen von Programmen begrenzter Funktionalität sowohl strukturell als auch funktional angemessen zu gestalten.
- kennen die grundlegenden Klassen und ihre Operationen, mit denen dateibezogene Operationen implementiert werden können.

Inhalt

- Grundkonzept der Programmiersprache Java

- Grundlegende Eigenschaften der Sprache
- Grundlegender Aufbau von Java-Programmen
- Ausführung von Java-Programmen
- Vorstellung der eingesetzten Entwicklungsumgebung (NetBeans)
- Grundlegende Programmelemente
 - Primitive Datentypen in Java
 - Variablen, Zuweisung, Gültigkeitsbereiche
 - Operatoren und Ausdrücke
 - Anweisungen
- Referenzdatentypen
 - Arrays
 - Klassen
- Statische Methoden
- Grundlegende Klassen
 - String
 - StringBuilder
 - Wrapper-Klassen für primitive Datentypen
 - Enum
- Grundkonzepte der Objektorientierung
 - Klassen und Instanzen mit Attributen und Methoden
 - Sichtbarkeit, Packages
 - Konstruktoren
 - Vererbung und Überschreiben
 - Dynamisches Binden, Polymorphie
 - Objektorientierte Realisierung rekursiver dynamischer Datenstrukturen (Listen)
 - Generische Typen
 - Abstrakte Klassen und Interfaces - Deklaration und Nutzung
 - Realisierung grafischer Benutzungsoberflächen
 - Behandlung von Laufzeitfehlern
 - Klassen zur Realisierung von Dateioperationen

Literatur

- ULLENBOOM, Christian:
Java ist auch eine Insel. 15. Auflage, Rheinwerk Verlag, 2020
- HABELITZ, Hans-Peter:
Programmieren lernen mit Java. 6. Auflage, Rheinwerk Computing, 2020
- SOLYMOSI, Andreas, GRUDE, Ulrich:
Grundkurs Algorithmen und Datenstrukturen in JAVA: Eine Einführung in die praktische Informatik. 6. Auflage, Springer Vieweg, 2017
- RATZ, Dietmar:
Grundkurs Programmieren in Java. 7. Auflage,
Hanser Verlag, 2014
- ABTS, Dietmar:
Grundkurs Java, Von den Grundlagen bis zu Datenbank- und Netzanwendungen, Springer Vieweg, 2015
- STEYER, Ralph:
Einführung in JavaFX: Moderne GUIs für RIAs und Java-Applikationen.
Springer-Vieweg, 2014
- EPPLE, Anton:
JavaFX 8: Grundlagen und fortgeschrittene Techniken. dpunkt.verlag, 2015
- SALTER, David; DANTAS, Rhawi:
NetBeans IDE 8 Cookbook. Packt Publishing, 2014

I.1.8.2 Übg. Programmstrukturen 2

| | |
|---------------------------------|--|
| Lehrveranstaltung | Übg. Programmstrukturen 2 |
| Dozent(en) | Gerit Kaleck |
| Hörtermin | 2 |
| Häufigkeit | jedes Semester |
| Lehrform | Übung/Praktikum/Planspiel |
| Semesterwochenstunden | 2 |
| ECTS | 2.0 |
| Prüfungsform | Abnahme |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, studentische Arbeit am Rechner |

Lernziele

Die Studierenden erwerben in der Übung die ...

- Fähigkeit zum praktischen Einsatz der Basiskonzepte objektorientierter Programmiersprachen sowie ihrer Umsetzung in Java.
- Fähigkeit zum Aufbau einfacher dynamischer Datenstrukturen im Kontext einer objektorientierten Programmiersprache und Fähigkeit der Anwendung grundlegender Algorithmen auf diesen Datenstrukturen.
- Fähigkeit zur Nutzung einer aktuellen, verbreiteten Entwicklungsumgebung.
- Fähigkeit zur Realisierung von vollständigen Software-Systemen kleineren Umfangs ausgehend von einer verbalen Aufgabenstellung.
- Fähigkeit zur Software-Entwicklung im kleinen Team.
- Fähigkeit zur Ermittlung geeigneter Testfälle zur Qualitätssicherung.
- Kenntnis der Grundregeln zur Gestaltung benutzungsgerechter Oberflächen und bedienfreundlicher Software.

Inhalt

- Einführung in die Programmierung mit Java und die Entwicklungsumgebung.
- Anwendung der in der Vorlesung vorgestellten Grundkonzepte der objektorientierten Programmierung durch das Lösen verbal formulierter Aufgabenstellungen in kleinen Teams.
- Testen und Präsentieren der sauber strukturierten Lösung.

Literatur

- HABELITZ, Hans-Peter: Programmieren lernen mit Java: Der leichte Einstieg für Programmieranfänger, Rheinwerk Computing, 2017 (5. Auflage) ISBN-13: 978-3836256056
- KOFLER, Michael: Java: Der kompakte Grundkurs mit Aufgaben und Lösungen im Taschenbuchformat, Rheinwerk Computing, 2019 (3. Auflage) ISBN-13: 978-3836269582

- ULLENBOOM, Christian: Java ist auch eine Insel, Rheinwerk Computing, 2020 (15. Auflage) ISBN-13: 978-3836277372

I.1.9 Programmierpraktikum

B036 Programmierpraktikum

| | |
|---------------------------------|--|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B036 |
| Bezeichnung | Programmierpraktikum |
| Lehrveranstaltung(en) | B036a Programmierpraktikum |
| Verantwortliche(r) | Dipl.-Ing. (FH) Gerit Kaleck |
| Zuordnung zum Curriculum | Computer Games Technology (Bachelor) E-Commerce (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Technische Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) |
| Verwendbarkeit | Das Modul baut auf die im Modul "Programmstrukturen 2" erworbenen Programmierkompetenzen auf. Es bildet die Grundlage für Module von Informatik-Studiengängen, in denen Programmierung von Softwareeinheiten größeren Umfangs und softwaretechnische Aspekte eine Rolle spielen, zum Beispiel die Module "Software-Design", "Software-Projekt" und "Softwarequalität". |
| Semesterwochenstunden | 0 |
| ECTS | 5.0 |
| Voraussetzungen | Kenntnis der zentralen Konzepte der Objektorientierten Programmierung und der Programmiersprache Java, Grundkenntnisse in der Benutzung einer Entwicklungsumgebung. |
| Dauer | 1 |

Lernziele

Die Studierenden verfügen nach Abschluss des Moduls über die Fähigkeit, aus einer textuellen, problemorientiert formulierten Aufgabenstellung die wesentlichen funktionalen Anforderungen an eine zu entwickelnde Software abzuleiten und in Form eines Pflichtenheftes zu dokumentieren. Sie sind zur Konzeption, insbesondere zur angemessenen Modularisierung von Softwaresystemen mittleren Umfangs in der Lage. Sie können die objektorientierte Programmiersprache Java einsetzen, um ein solches Softwaresystem eigenständig objektorientiert zu implementieren. Sie sind fähig, dabei eine moderne Entwicklungsumgebung zu nutzen.

Dabei verfügen sie über das Problembewusstsein im Hinblick auf die benutzungsgerechte Softwaregestaltung und verfügen über Kenntnisse hinsichtlich der Grundregeln von benutzungsgerechter Oberflächengestaltung. Sie verfügen zudem über Fähigkeiten zum angemessenen Einsatz von Interaktionselementen in grafischen Oberflächen bei Aufgabenstellungen mittleren Schwierigkeitsgrades.

Sie verfügen über Basiskenntnisse hinsichtlich der Qualitätssicherung von Software in Form einfacher Teststrategien und können diese einsetzen, um die funktionale Korrektheit und ein ausreichendes Maß an Zuverlässigkeit der Software zu gewährleisten.

Die Studierenden besitzen die Fähigkeit, die Entwicklung einer Software und die Software selbst in einer geeigneten Form zu dokumentieren und zu präsentieren.

I.1.9.1 Programmierpraktikum

| | |
|---------------------------------|--------------------------------|
| Lehrveranstaltung | Programmierpraktikum |
| Dozent(en) | Gerit Kaleck |
| Hörtermin | 2 |
| Häufigkeit | jedes Semester |
| Lehrform | Übung/Praktikum/Planspiel |
| Semesterwochenstunden | 0 |
| ECTS | 5.0 |
| Prüfungsform | Praktikumsbericht / Protokoll |
| Sprache | deutsch |
| Lehr- und Medienform(en) | studentische Arbeit am Rechner |

Lernziele

Die Studierenden erwerben mit diesem Praktikum die ...

- Fähigkeit zum praktischen Einsatz der Basiskonzepte objektorientierter Programmiersprachen sowie ihrer Umsetzung in Java in einem Softwareprojekt mittleren Umfangs.
- Fähigkeit zur Nutzung einer aktuellen, verbreiteten Entwicklungsumgebung.
- Fähigkeit zur eigenständigen Strukturierung und Realisierung von vollständigen Softwaresystemen mittleren Umfangs ausgehend von einer problemorientierten Aufgabenstellung.
- Fähigkeit zur Softwareentwicklung und -dokumentation.
- Fähigkeit zur Anwendung der Grundregeln benutzungsgerechter Oberflächengestaltung.

Inhalt

- Entwicklung eines vollständigen Softwaresystems mittleren Umfangs in Java ausgehend von einer problemorientierten Aufgabenstellung.
- Strukturierung und Modularisierung des Projektes.
- Eigenständiger Entwurf passender Datenmodelle.
- Benutzungsgerechte Gestaltung der Oberfläche.
- Testen der entstandenen Software und Dokumentation der Tests.
- Erstellen einer Programmdokumentation und eines Benutzerhandbuchs.

Literatur

- ULLENBOOM, Christian:
Java ist auch eine Insel
Rheinwerk Computing, 2020 (15. Auflage)
ISBN-13: 978-3836277372
- EPPLE, Anton:
JavaFX 8: Grundlagen und fortgeschrittene Techniken

dpunkt Verlag, 2015
ISBN-13: 978-3864901690

- ZÖRNER, Stefan:
Softwarearchitekturen dokumentieren und kommunizieren: Entwürfe, Entscheidungen
und Lösungen nachvollziehbar und wirkungsvoll festhalten
Carl Hanser Verlag, 2015 (2. Auflage)
ISBN-13: 978-3446443488

I.1.10 Formale Sprachen

B232 Formale Sprachen

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B232 |
| Bezeichnung | Formale Sprachen |
| Lehrveranstaltung(en) | B232a Formale Sprachen |
| Verantwortliche(r) | B.Sc. Marco Pawlowski |
| Zuordnung zum Curriculum | Computer Games Technology (Bachelor) Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) |
| Verwendbarkeit | Das Modul baut auf dem Modul "Mathematische Konzepte und Diskrete Mathematik" auf. Es liefert ein tieferes Verständnis von Programmiersprachen und ihrer Ausführung auf Maschinen und unterstützt damit sowohl grundlegende als auch fortgeschrittene Module zur Softwaretechnik wie "Programmstrukturen 1", "Programmstrukturen 2", "Algorithmen und Datenstrukturen", "Systemnahe Programmierung", "Fortgeschrittene Objektorientierte Programmierung" und "Software-Design". |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Die Studierenden müssen elementare Logik und Mengenlehre sowie darauf aufbauend Relationen, Funktionen und Graphen beherrschen. Ferner wird eine gute Beherrschung elementarer Beweisprinzipien vorausgesetzt. Die Teilnehmer sollten außerdem die Grundlagen von imperativen Programmiersprachen und ihre typischen Aspekte wie Variablen, Typisierung, Ausdrücke, Anweisungen und Kontrollstrukturen kennengelernt haben. |
| Dauer | 1 |

Lernziele

Nach Abschluss des Moduls können die Studierenden umgangssprachliche Sachverhalte sicher in eine logische Formelsprache übertragen. Sie beherrschen die Begriffe und Modelle von Automaten, formalen Sprachen und Algorithmen.

Sie können diese Begriffe mit Beispielen belegen und vorgegebene Beispiele richtig einordnen.

Die Studierenden können den Zusammenhang zwischen formalen Sprachen und Automaten bestimmten Typs benennen und an Beispielen belegen. Sie verstehen die grundlegende Bedeutung der Berechenbarkeit und Komplexität von Problemen für die Praxis und können die wesentlichen Zusammenhänge an Beispielen benennen.

Sie kennen die Begriffswelt von Compilern und Interpretierern und können die Abläufe im Rahmen der Übersetzung von Programmen per Compiler grob erläutern. Ihnen ist die Verwendbarkeit von Techniken des Compilerbaus in allgemeinen Softwaresystemen klar und sie können einfache domänenspezifische Sprachen per regulärer Ausdrücke und kontextfreier Grammatiken definieren sowie mittels Scanner- und Parser-Generatoren in ihre Softwareprojekte integrieren.

I.1.10.1 Formale Sprachen

| | |
|---------------------------------|--|
| Lehrveranstaltung | Formale Sprachen |
| Dozent(en) | Marco Pawlowski |
| Hörtermin | 2 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, Handout, interaktive Entwicklung und Diskussion von Modellen, Softwaredemonstration, Tafel |

Lernziele

Die Studierenden ...

- unterscheiden verschiedene Arten von Grammatiken und Sprachklassen.
- nennen die Eigenschaften abstrakter Automaten.
- erkennen den Zusammenhang zwischen Automaten und Sprachen.
- nennen die Grundbegriffe der Programmsyntax und von Compilern.
- beschreiben den Übersetzungsprozess formaler Sprachen.
- nennen und charakterisieren die Phasen des Übersetzungsprozesses.
- entwerfen einfache domänenspezifische formale Sprachen und definieren sie mittels regulärer Ausdrücke und kontextfreier Grammatiken.
- nutzen Scanner- und Parsergeneratoren zur Verarbeitung einfacher domänenspezifischer Sprachen.
- erläutern die Transformation höherer Programmiersprachen in Maschinensprache.
- verstehen und differenzieren die Arbeitsweise von Compilern und Interpretierern.
- verstehen Funktionsweise und Anwendungsbereiche virtueller Maschinen.

Inhalt

- Endliche Automaten
 - Endliche Automaten mit und ohne Ausgabe
 - Deterministische und nichtdeterministische Automaten
 - Reguläre Sprachen
 - Reguläre Ausdrücke
 - Äquivalenz und Minimierung endlicher Automaten

- Anwendungen endlicher Automaten
- Pumping-Lemma für reguläre Sprachen
- Nicht-reguläre Sprachen
- Formale Sprachen
 - Ersetzungssysteme
 - Grammatiken
 - Chomsky-Hierarchie
 - Kontextfreie Grammatiken
 - Normalformen
 - Pumping-Lemma für kontextfreie Sprachen
 - Abschlusseigenschaften kontextfreier Sprachen
 - Entscheidbarkeit bei kontextfreien Sprachen
 - Eindeutigkeit und Mehrdeutigkeit von Grammatiken
 - Syntaxanalyse von Programmiersprachen
 - Kellerautomaten
 - Kontext-sensitive Sprachen
 - Linear-beschränkte Automaten
 - Turingmaschinen
- Compiler im Überblick
 - Compilerphasen
 - Compiler und Interpretierer
 - Anwendungsfelder
- Lexikalische Analyse
 - Nutzung regulärer Ausdrücke und nicht-deterministischer bzw. deterministischer endlicher Automaten
 - Scanner und Scanner-Generatoren
- Syntaxanalyse
 - Rekursiver Abstieg
 - LL- und LR-Parser
 - Parser-Generatoren
- Semantische Analyse
 - Typüberprüfung
- Codeerzeugung und -optimierung

- Virtuelle Maschinen

Literatur

- Roland Backhouse: Programmkonstruktion und Verifikation, Hanser 1989, ISBN 3-446-15056-0, Englische Neuauflage: Program Construction: Calculating Implementations from Specifications, Wiley 2003, ISBN 0470848820
- David Harel / Yishai Feldman: Algorithmen, Springer 2006, ISBN 3-540-24342-9
- John E. Hopcroft / Rajeev Motwani / Jeffrey D. Ullman: Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie. 2. überarb. Aufl. München: Addison-Wesley Longman Verlag, 2002, ISBN 978-3827370204
- Michael Huth / Mark Ryan: Logic in Computer Science, Cambridge University Press 2004 (2. Auflage), ISBN 052154310X
- Rainer Lang: Theoretische Informatik. Vorlesungsskript, FH-Wedel, 2008
- Uwe Schöning: Logik für Informatiker, Spektrum 2000 (5. Auflage), ISBN 3-8274-1005-3
- Gottfried Vossen / Kurt-Ulrich Witt: Theoretische Informatik, Verlag Vieweg & Teubner 2004 (3. Auflage), ISBN 978-3528231477
- Wegener, Ingo: Theoretische Informatik - eine algorithmenorientierte Einführung. 2. Aufl. Stuttgart: B., G. Teubner Verlag, 1999, ISBN 978-3519121237
- Aho, Alfred V.; Sethi, Ravi; Ullman, Jeffrey D.: Compilers, Principles, Techniques and Tools, Addison-Wesley Longman, Amsterdam, 2000 (Repr.)
- Appel, Andrew W.; Palsberg, Jens: Modern Compiler Implementation in Java, 2.nd edition, Cambridge University Press, 2002, ISBN: 0-521-82060-X
- Wirth, Niklaus: Grundlagen und Techniken des Compilerbaus, Oldenbourg, 1995, ISBN: 3-486-24374-8

I.1.11 Rechnerstrukturen und Digitaltechnik

B023 Rechnerstrukturen und Digitaltechnik

| | |
|---------------------------------|--|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B023 |
| Bezeichnung | Rechnerstrukturen und Digitaltechnik |
| Lehrveranstaltung(en) | B023a Digitaltechnik 2 B023a Rechnerstrukturen |
| Verantwortliche(r) | Prof. Dr. Sergei Sawitzki |
| Zuordnung zum Curriculum | IT-Ingenieurwesen (Bachelor) Informatik (Bachelor) Smart Technology (Bachelor) Technische Informatik (Bachelor) |
| Verwendbarkeit | Das Modul "Rechnerstrukturen und Digitaltechnik" baut auf den im Modul "Einführung in Digitaltechnik" erworbenen Kenntnissen und Fähigkeiten auf. Die im Modul "Rechnerstrukturen und Digitaltechnik" erworbenen Kompetenzen stellen die Grundlagen für zum Beispiel die Module "Diskrete Systeme" und "Systementwurf mit VHDL" dar. Das Modul kann sinnvoll mit den Modulen, die einerseits Grundlagen der Digitaltechnik beleuchten und andererseits ein Rechnersystem auf höheren Abstraktionsebenen (über dem Gatterniveau) behandeln, kombiniert werden. Das Modul ist fachübergreifend für alle Studiengänge relevant, die eine grundlegende Hardware-Kompetenz sowie Kenntnisse moderner Rechnerarchitekturen voraussetzen und hat somit eine direkte Verbindung zu den Studiengangszielen des Studiengangs "Bachelor Technische Informatik". |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Es werden Kompetenzen aus dem Modul "Einführung in Digitaltechnik" vorausgesetzt. |
| Dauer | 1 |

Lernziele

Aufbauend auf den im Modul "Einführung in Digitaltechnik" erworbenen Kompetenzen bekommen die Studierenden das Verständnis für zustandsbehaftete Systeme und deren Implementierung in Form von Schaltwerken vermittelt. Die Abstraktion mathematischer Darstellung

von Zustandsautomaten ist nach dem Absolvieren des Moduls mit ihren technischen Abbildern versehen. Dabei lernen die Studierenden zu begreifen, worin die Unterschiede zwischen Modellen und realen Schaltungen und Systemen bestehen, warum Abstraktionen und modellhafte Darstellungen unvermeidlich sind und wo deren Grenzen liegen. Aufbauend auf den einfacheren Schaltungen werden Rechnersysteme als komplexe Vertreter digitaler Systeme betrachtet. Die Studierende lernen den Aufbau und die Funktionsweise moderner Rechner kennen und mit den Begriffen und Konzepten aus dem Bereich Rechnerarchitektur sicher umzugehen. Sie sollten in die Lage versetzt werden, Abläufe in Hardware eines modernen Rechners zu begreifen und klassische sowie innovative Architekturkonzepte zu erkennen und richtig einzuordnen. Ein wesentliches Lernziel besteht außerdem in Erkennung der Bedeutung der Zeitverhaltens von einfachen logischen Gattern und Schaltungen und deren Einflusses auf die Leistungsfähigkeit digitaler Systeme (Verzögerungs-, Setz und Haltezeiten, Taktfrequenz, Steigerung des Durchsatzes in modernen Rechnerarchitekturen).

I.1.11.1 Digitaltechnik 2

| | |
|---------------------------------|------------------------------------|
| Lehrveranstaltung | Digitaltechnik 2 |
| Dozent(en) | Sergei Sawitzki |
| Hörtermin | 2 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 2 |
| ECTS | 2.5 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, Handout, Tafel |

Lernziele

Die Studierenden ...

- ergänzen und erweitern die Grundlagen aus der Lehrveranstaltung Digitaltechnik 1
- erlangen Kenntnisse über Aufbau und Funktionsweise von Speicherelementen und Schaltwerken
- erlernen die Methoden zur Analyse, Darstellung und Vereinfachung von endlichen Zustandsautomaten
- begreifen ein Schaltwerk als technische Umsetzung eines endlichen Zustandsautomaten
- gewinnen eine Einsicht in die Methoden der Zeitverhaltensanalyse und Zeitverhaltensoptimierung von digitalen Systemen, sowie lernen das Zeitverhalten und die Zeitvorgaben beim Entwurf digitaler Systemen zu berücksichtigen;
- erlangen die Fähigkeit, digitale Systeme in der Gesamtheit verschiedener Aspekte zu begreifen, die für ihren praktischen Einsatz eine Rolle spielen (Schnittstellen, Komplexität, Zeitverhalten, Leistungsaufnahme, usw.)
- erlangen die Fähigkeit, digitale Systeme mittlerer Komplexität zu begreifen, zu spezifizieren, zu entwerfen und zu optimieren.

Inhalt

- Einleitung
 - Lernziele
 - Organisatorisches
 - Literatur
 - Bezeichnungen und Konventionen
 - Einordnung und historische Entwicklung
 - Voraussetzungen
- Schaltwerke

- Einleitung und Grundbegriffe, Definitionen
- Speicherelemente
- Analyse
- Synthese
- Zusammenschaltung
- Transformationen
- Zustandskodierung
- Zustandsminimierung
- Realisierung, Beispiele
- Zeitverhalten
 - Zeitverhalten von Schaltnetzen
 - Modellierung der Gatter- und Leitungsverzögerungen
 - Statische Timing-Analyse (STA)
 - Zeitverhalten von Schaltwerken
 - Metastabilität

Literatur

- Hoffmann, Dirk: Grundlagen der technischen Informatik, Carl Hanser Verlag 2007
- Schiffmann, Wolram; Schmitz, Robert: Technische Informatik, in 3 Bänden. 3. Auflage Springer Verlag, 1996
- Veendrick, Harry: Nanometer CMOS ICs, Springer 2008
- Rabaey, Jan; Chandrakasan, Anantha; Nikolic, Borivoje: Digital Integrated Circuits, A Design Perspective, 2nd edition, Prentice Hall 2003
- Beuth, Klaus: Elektronik 4. Digitaltechnik, 13. Auflage, Vogel Verlag und Druck 2003

I.1.11.2 Rechnerstrukturen

| | |
|---------------------------------|-------------------|
| Lehrveranstaltung | Rechnerstrukturen |
| Dozent(en) | Dennis Säring |
| Hörtermin | 2 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 2 |
| ECTS | 2.5 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | E-Learning |

Lernziele

Die Studierenden ...

- kennen grundlegende Rechnerarchitekturkonzepte, die beschreiben, wie verschiedene Baugruppen von Rechnern zusammenarbeiten und wie sich unterschiedliche Rechner-systeme voneinander unterscheiden
- haben die Funktionselemente von Rechnern mit ihren typischen Systemeigenschaften und deren Abbildung auf ein bestimmtes Architekturmodell gelernt
- können das Zusammenwirken der beteiligten Hardware- und Softwarekonzepte im Rahmen einer Aufgabe zur Informationsverarbeitung einschätzen
- besitzen ein Verständnis für Ansätze zur Steigerung der Systemleistung insbesondere unter Berücksichtigung der Aspekte von Parallelität

Inhalt

- Entwicklung der Rechnerarchitekturen
- Grundlegende Rechnerstrukturen
- Architekturkonzepte
- Mikroprogrammierung
- Moderne parallele und nicht-sequentielle Architekturkonzepte
- Mehrprozessorsysteme
- Aktuelle und zukünftige Entwicklungen

Literatur

- Märtin: Einführung in die Rechnerarchitektur, Fachbuchverlag Leipzig, 2003
- Oberschelp, Gossen: Rechneraufbau und Rechnerstrukturen, Verlag Oldenbourg 1998
- van de Goor: Computer Architecture and Design, Verlag Addison Wesley, 1989
- Müller-Schloer, Schmitter: RISC-Workstation Architekturen, Verlag Springer 1991

- Ungerer: Datenfluß-Rechner, Verlag Teubner, 1993

I.1.12 Deskriptive Statistik und Grundlagen der Linearen Algebra

B019 Deskriptive Statistik und Grundlagen der Linearen Algebra

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B019 |
| Bezeichnung | Deskriptive Statistik und Grundlagen der Linearen Algebra |
| Lehrveranstaltung(en) | B019a Grundlagen der Linearen Algebra B019a Deskriptive Statistik |
| Verantwortliche(r) | Prof. Dr. Franziska Bönte |
| Zuordnung zum Curriculum | Betriebswirtschaftslehre (Bachelor) Computer Games Technology (Bachelor) Data Science & Artificial Intelligence (Bachelor) E-Commerce (Bachelor) IT-Ingenieurwesen (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Smart Technology (Bachelor) Technische Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) Wirtschaftsingenieurwesen (Bachelor) |
| Verwendbarkeit | Das Modul "Deskriptive Statistik & Grundlagen der Linearen Algebra" ist ein Einführungsmodul. Zusammen mit dem Modul "Analysis", stellt es die Grundlage für nahezu alle quantitativ ausgerichteten weiter führenden Module und Veranstaltungen des Studienverlaufs dar. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Vorausgesetzt wird ein solides Schulwissen der Mathematik und mindestens durchschnittliche mathematische Begabung. |
| Dauer | 1 |

Lernziele

In diesem Modul werden grundlegende mathematische Kenntnisse aus den Bereichen lineare Algebra und deskriptiven Statistik, wie sie als Grundlage für ein quantitativ ausgerichtetes

Studium unerlässlich sind, vermittelt. Der Fokus liegt auf der Vektor- und Matrizenrechnung, linearen Gleichungssystemen und statistischer Datenanalyse.

Die Lernenden sind in der Lage ausgewählte Problemstellungen aus den Bereichen Technik, Naturwissenschaft, Ökonomie und Informatik, mittels der im Modul vermittelten mathematischen und statistischen Methoden zu modellieren und analysieren. Die Lernenden können für die vermittelten Inhalte praxisrelevante Anwendungsbeispiele benennen. Die Lernenden können eigenständig Lösungsmethoden für ausgewählte Problemstellungen auswählen, die Lösungsmethodik bis zum Ergebnis durchführen und die erhaltenen Ergebnisse kritisch bewerten.

I.1.12.1 Grundlagen der Linearen Algebra

| | |
|---------------------------------|---------------------------------|
| Lehrveranstaltung | Grundlagen der Linearen Algebra |
| Dozent(en) | Andreas Haase |
| Hörtermin | 2 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 2 |
| ECTS | 2.5 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Handout, Overheadfolien, Tafel |

Lernziele

Nach dem erfolgreichen Besuch der Vorlesung sind die Lernenden in der Lage ...

- lineare algebraische Gleichungssysteme mittels des Gauß-Algorithmus in die Lösbarkeitskategorien (eindeutig lösbar, unendlich viele Lösungen, unlösbar) einzuteilen und ggfs. die Lösung anzugeben.
- die Techniken und Methoden der Vektorrechnung anzuwenden.
- die Techniken und Methoden der Matrixrechnung anzuwenden.
- die Determinante einer niedrigdimensionalen Matrix zu berechnen und den Zusammenhang der Determinante zur Lösungstheorie linearer Gleichungssysteme herzustellen
- einfache technische oder ökonomische Systeme mittels der Techniken und Methoden der linearen Algebra zu modellieren und aus der ermittelten Lösung der mathematischen Formulierung das System quantitativ zu beurteilen.

Inhalt

- Lineare algebraische Gleichungssysteme
 - Gauß-Algorithmus
 - Systematisierung des Lösungsverhaltens
 - Unterbestimmte Systeme
- Matrixrechnung
 - Matrixalgebra
 - Inverse Matrix
 - Matrixgleichungen
 - Zusammenhang mit linearen Gleichungssystemen
- Determinanten
 - Definition

- Zusammenhang mit linearen Gleichungssystemen
- Vektorrechnung
 - Geometrische Vektoren
 - Rechenregeln
 - Lineare (Un-)Abhängigkeit
 - Rang einer Matrix
 - Nochmal Gleichungssysteme, Rangkriterium

Literatur

- PAPULA, Lothar:
Mathematik für Ingenieure und Naturwissenschaftler,
Band 2, Teil I. 13. Aufl. Wiesbaden: Vieweg + Teubner Verlag 2012
- HELM, Werner; PFEIFER, Andreas; OHSER, Joachim:
Mathematik für Wirtschaftswissenschaftler.
1. Aufl. München: Carl Hanser Verlag 2011
- GRAMLICH, Günter:
Lineare Algebra: Eine Einführung.
1. Aufl. München: Carl Hanser Verlag 2011
- TESCHL, Gerald; TESCHL, Susanne:
Mathematik für Informatiker,
Band 1: Diskrete Mathematik und lineare Algebra.
3. Aufl. Heidelberg: Springer Verlag 2008
- FISCHER, Gerd:
Lineare Algebra: Eine Einführung für Studienanfänger.
18. aktualisierte Aufl. Wiesbaden: Springer Verlag 2014

I.1.12.2 Deskriptive Statistik

| | |
|---------------------------------|---|
| Lehrveranstaltung | Deskriptive Statistik |
| Dozent(en) | Franziska Bönte |
| Hörtermin | 2 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 2 |
| ECTS | 2.5 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, E-Learning, Gastreferenten, interaktive Entwicklung und Diskussion von Modellen, Online-Aufbereitung, Tafel, Tutorien |

Lernziele

Nach der Lehrveranstaltung können die Studierenden ...

- Statistische Daten verdichten und graphisch aussagekräftig darstellen
- Wesentliche Aussagen über Daten anhand geeigneter Kennzahlen treffen und interpretieren
- Die Ableitung von Regressionsformeln verstehen und komplexe Regressions- und deskriptive Zeitreihenanalysen abgestimmt auf den jeweiligen Datensatz durchführen und interpretieren
- sicher im Umgang mit Meß- und Indexzahlen agieren

Inhalt

Im Rahmen der beschreibenden / deskriptiven Statistik werden folgende Themen behandelt:

- Begrifflichkeiten
- Lage- und Streuungsmaße
- Abhängigkeitsmessung bei qualitativen, komperativen und quantitativen Merkmalen insbesondere Regressionsanalyse
- Deskriptive Zeitreihenanalyse mit Trend-, Saison- und Restkomponentenschätzung nach unterschiedlichen Methoden
- Meß- und Indexzahlen

Literatur

- Christensen, B.; Christensen, S.; Missong, M.: Statistik klipp & klar; 2019; Springer Gabler Verlag
- Bamberg, G.; Baur, F; Krapp, M: Statistik; 18. Auflage; 2017; De Gruyter Oldenbourg Verlag; München

- Missong, Martin; Aufgabensammlung zur deskriptiven Statistik; 2005; 7. Auflage; Verlag R. Oldenbourg, München.
- Schneider, Wolfgang; Kornrumpf, J.; Mohr, Walter; Statistische Methodenlehre --
- Definitions- und Formelsammlung zur deskriptiven und induktiven Statistik mit Erläuterungen; 1993; Verlag Oldenbourg, München.

I.1.13 Datenschutz und Wirtschaftsprivatrecht

B042 Datenschutz und Wirtschaftsprivatrecht

| | |
|---------------------------------|--|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B042 |
| Bezeichnung | Datenschutz und Wirtschaftsprivatrecht |
| Lehrveranstaltung(en) | B042a Datenschutz B042a Wirtschaftsprivatrecht |
| Verantwortliche(r) | Prof. Dr. Gerd Beuster |
| Zuordnung zum Curriculum | Betriebswirtschaftslehre (Bachelor) Data Science & Artificial Intelligence (Bachelor) IT-Ingenieurwesen (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) Wirtschaftsingenieurwesen (Bachelor) |
| Verwendbarkeit | Keine. |
| Semesterwochenstunden | 6 |
| ECTS | 5.0 |
| Voraussetzungen | Das Modul setzt ein grundlegendes Verständnis des deutschen Rechtssystem und seiner Begrifflichkeiten sowie elementare Kenntnisse über die Grundprinzipien deutscher Gesetzgebung voraus. |
| Dauer | 1 |

Lernziele

Wirtschaftliche Vorgänge sind in einem rechtlichen Ordnungsrahmen eingebettet. Nach Abschluss des Moduls kennen die Studierenden diese grundlegenden rechtlichen Anforderungen für ausgewählte Bereiche. Hierdurch sind die Studierenden in der Lage, rechtliche Anforderungen in typische Wirtschaftsvorgänge einzuordnen und zu verstehen. Sie sind somit fähig, Praxisfälle rechtlich zutreffend einzuordnen und unter Heranziehung einschlägiger Rechtsnormen zu würdigen. Die Studierenden sind ferner fähig, in ihrem späteren Wirkungskreis (datenschutz- und wirtschafts-) rechtliche Fragestellungen einzuordnen, um bei Bedarf auf Spezialistenunterstützung gezielt zurückgreifen zu können.

I.1.13.1 Datenschutz

| | |
|---------------------------------|--------------|
| Lehrveranstaltung | Datenschutz |
| Dozent(en) | Steffen Weiß |
| Hörtermin | 4 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 2 |
| ECTS | 2.0 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | |

Lernziele

Die Studierenden sind fähig, in ihrem späteren Wirkungskreis datenschutzrechtliche Fragestellungen einzuordnen, um bei Bedarf auf Spezialistenunterstützung gezielt zurückgreifen zu können.

Inhalt

- Gesetzliche Grundlagen des Datenschutzes
 - Anwendung und praktische Umsetzung des Bundesdatenschutzgesetzes (BDSG)
 - Wesentliche Grundlagen aus ausgewählten bereichsspezifischen und bereichsübergreifenden Datenschutzgesetzen
 - Rechte, Pflichten und Aufgaben des betrieblichen Datenschutzbeauftragten zur Einrichtung des Datenschutzmanagements
 - Datenschutz in der Werbepaxis
- Technisch-organisatorischer Datenschutz
 - Grundanforderungen und Grundfunktionen der IT-Sicherheit in Bezug auf die Anforderungen der Datenschutzgesetze
 - Risikomanagement und Schlüsseltechnologien zur Realisierung des technisch-organisatorischen Datenschutzes
 - Kosten-/Nutzen des Datenschutzes
 - Verfahren zur Umsetzung des gesetzlichen Anforderungen des technisch-organisatorischen Datenschutzes
 - Auswahlverfahren zu geeigneten und angemessenen IT-Sicherheitsmechanismen

Literatur

- Bundesdatenschutzgesetz (BDSG) vom Januar 2003, novelliert im Juli 2009
- Koch (Hrsg.):
Handbuch des betrieblichen Datenschutzbeauftragten. 4. Aufl. Frechen: Datakontext

- MÜNCH, Peter:
Technisch-organisatorischer Datenschutz. 4. Aufl. Frechen: Datakontext, 2010

I.1.13.2 Wirtschaftsprivatrecht

| | |
|---------------------------------|------------------------|
| Lehrveranstaltung | Wirtschaftsprivatrecht |
| Dozent(en) | Felix Reiche |
| Hörtermin | 4 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 4 |
| ECTS | 3.0 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | |

Lernziele

Die Studierenden können einfache Sachverhalte des Zivilrechts rechtlich zutreffend einordnen und unter Heranziehung einschlägiger Gesetzestexte würdigen.

Inhalt

- Zivilrecht
- BGB
 - Allgemeiner Teil
 - Recht der Schuldverhältnisse
 - Sachenrecht
- HBG
 - Handelsstand
 - Handelsgeschäfte
- Gesellschaftsrecht
 - Personenhandelsgesellschaften
 - Juristische Personen
- Wettbewerbsrecht / Arbeitsrecht / Prozeßrecht werden fragmentarisch mit bearbeitet

Literatur

- MÜSSIG, Peter:
Wirtschaftsprivatrecht.
17. Aufl. Heidelberg: C.F. Müller, 2014.
- Bürgerliches Gesetzbuch.
- Handelsgesetzbuch.

I.1.14 Algorithmen und Datenstrukturen

B040 Algorithmen und Datenstrukturen

| | |
|---------------------------------|--|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B040 |
| Bezeichnung | Algorithmen und Datenstrukturen |
| Lehrveranstaltung(en) | B040a Algorithmen und Datenstrukturen B040b Übg. Algorithmen & Datenstrukturen |
| Verantwortliche(r) | M.Sc. Christian Uhlig |
| Zuordnung zum Curriculum | Computer Games Technology (Bachelor) Data Science & Artificial Intelligence (Bachelor) E-Commerce (Bachelor) IT-Ingenieurwesen (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Smart Technology (Bachelor) Technische Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) |
| Verwendbarkeit | Das Modul setzt unmittelbar auf den Inhalten des Moduls „Programmstrukturen 2“ auf und eignet sich damit als Weiterqualifikation im Anschluss an „Programmstrukturen 2“ und das „Programmierpraktikum“. Es kann ergänzend mit fortgeschrittenen Modulen zur Software-Technik kombiniert werden, insbesondere mit „Software-Design“, „Fortgeschrittene Objektorientierte Programmierung“ und „Systemnahe Programmierung“. |
| Semesterwochenstunden | 5 |
| ECTS | 5.0 |
| Voraussetzungen | Vorausgesetzt werden theoretische und praktische Grundkenntnisse in der Programmiersprache Java, die insbesondere auch die Abbildung abstrakter Datentypen per Interfaces und abstrakter Klassen und die Verwendung generischer Typen umfassen. |
| Dauer | 1 |

Lernziele

Nach Abschluss des Moduls sind sich die Studierenden der Einflussfaktoren auf den Speicherbedarf einer Datenstruktur und auf die Laufzeit der darauf ausgeführten Algorithmen bewusst. Sie können Laufzeit und Speicherbedarf systematisch auf Ebene des Programmcodes und aufgrund empirischer Untersuchung analysieren und vergleichend beurteilen, wobei sie zwischen dem konstanten Faktor und dem Laufzeitwachstum in Abhängigkeit von der Problemgröße unterscheiden. Hierbei verstehen und nutzen sie Komplexitätsklassen insbesondere per Groß-O-Notation und können eigenen Programmcodes in typische Komplexitätsklassen einordnen.

Weiterhin sind sich die Studierenden der Bedeutung des Sortierens und Suchens in der Softwaretechnik bewusst und kennen die wesentlichen theoretischen Grundlagen dieser Probleme und entsprechender Algorithmen.

Sie können die Trennung in abstrakte Datentypen und zugehörige Implementierungen erläutern und gewinnbringend für eigene Softwareprojekte anwenden. Hierbei wählen sie abhängig vom Anwendungsfall zwischen typischen abstrakten Datentypen wie z.B. Listen und Verzeichnissen aus.

Ihnen sind die Konzepte, die Funktionsweise wichtiger Operationen und die damit verbundenen Eigenschaften typischer Implementierungen wie z.B. verketteter Listen, Arrays und Binärer Suchbäume bekannt, so dass sie ausgehend von den einzusetzenden Operationen geeignete Implementierungen für den verwendeten abstrakten Datentypen auswählen.

Ausgehend von den erworbenen Grundlagenkenntnissen nutzen die Studierenden den Java Collections Framework und wählen dabei zielgerichtet abstrakte Datentypen und Implementierungen des Frameworks aus.

Die Studierenden können die Motivation und die Grundlagen nebenläufiger Algorithmen am Beispiel von dynamischem Multithreading erläutern und die Konzepte auf die Programmiersprache Java übertragen und im Rahmen einfacher Problemstellungen einsetzen.

I.1.14.1 Algorithmen und Datenstrukturen

| | |
|---------------------------------|---|
| Lehrveranstaltung | Algorithmen und Datenstrukturen |
| Dozent(en) | Christian Uhlig |
| Hörtermin | 3 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 3 |
| ECTS | 3.0 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, Handout, interaktive Entwicklung und Diskussion von Modellen, Softwaredemonstration |

Lernziele

Die Studierenden ...

- analysieren, diskutieren und vergleichen einfache Algorithmen und Datenstrukturen hinsichtlich ihres Bedarfs an Laufzeit und Speicher.
- differenzieren bei der Analyse von Algorithmen hinsichtlich best case, worst case und average case.
- differenzieren die Laufzeit von Algorithmen nach ihrem konstanten Faktor und ihrem Wachstum in Abhängigkeit von der Problemgröße.
- beurteilen die Laufzeit von Algorithmen ausgehend von Komplexitätsklassen in den Groß-O-, Groß-Omega- und Groß-Theta-Notationen.
- nennen und erläutern wesentliche Aspekte, Funktionsweisen und Eigenschaften von Algorithmen zum Suchen und Sortieren.
- erläutern die Differenzierung in abstrakte Datentypen und ihre Implementierung.
- nennen und erläutern typische abstrakte Datentypen wie Listen, Mengen, Verzeichnisse und Warteschlangen mit ihren Operationen und Anwendungsbereichen.
- nennen und erläutern Motivation, Funktionsweise und Eigenschaften typischer Implementierungen abstrakter Datentypen mit verketteten Listen, Arrays, Baumstrukturen und Hash-Tabellen.
- wählen zu einer gegebenen Problemstellung einen geeigneten abstrakten Datentypen nebst einer geeigneten Implementierung aus.
- wenden die abstrakten Datentypen und Implementierungen des Java Collections Frameworks an.
- erläutern die Grundkonzepte nebenläufiger Programmierung.
- wenden die Primitiven dynamischen Multithreadings bei der Formulierung von Programmen an.

Inhalt

- Analyse von Algorithmen
 - Laufzeit und Speicherbedarf
 - Groß-O / Groß-Omega / Groß-Theta Notationen
 - Amortisierte Laufzeitanalyse
 - Iterative vs rekursive Implementierungen
- Sortieren und Suchen
- Listenstrukturen
 - Verkettete Listen
 - Arraybasierte Listen
 - Skiplisten
- Baumstrukturen
 - Binäre Suchbäume
 - Balancierte Suchbäume: 2-3-Bäume
 - Balancierte Binäre Suchbäume: Rot/Schwarz-Bäume
 - Balancierte Binäre Suchbäume: AVL-Bäume
 - Spreizbäume
 - Tries
 - Arraybasierte Binäre Heaps
- Hash-Tabellen
- Abstrakte Datentypen und ihre Implementierung
 - Listen
 - Mengen
 - Verzeichnisse
 - Warteschlangen
- Java Collections Framework
- Nebenläufige Algorithmen

Literatur

- Uhlig, Christian: Algorithmen und Datenstrukturen, Vorlesungsunterlagen im Web: <http://www.fh-wedel.de/~uhl/aud.html>
- Sedgewick, Robert; Wayne, Kevin: Algorithms, 4th Edition, Addison-Wesley, 2011
- Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford: Introduction to Algorithms, 3rd Edition, The MIT Press, 2009

- Knuth, Donald E.: The Art of Computer Programming Vol. 1 Fundamental Algorithms, 3rd Edition, Addison-Wesley, 1997
- Knuth, Donald E.: The Art of Computer Programming Vol. 3 Sorting and Searching, 2nd Edition, Addison-Wesley, 1998
- Wirth, Niklaus: Algorithmen und Datenstrukturen, 5. Auflage, Teubner, 2013
- Aho, Alfred V.; Hopcroft, John E.; Ullman, Jeffrey D.: The Design and Analysis of Computer Algorithms, 1st Edition, Pearson, 1975
- Aho, Alfred V.; Hopcroft, John E.; Ullman, Jeffrey D.: Data Structures and Algorithms, Addison-Wesley, 1983
- Aho, Alfred V.; Ullman, Jeffrey D.: Foundations of computer science, Computer Science Press, 1992

I.1.14.2 Übg. Algorithmen & Datenstrukturen

| | |
|---------------------------------|--|
| Lehrveranstaltung | Übg. Algorithmen & Datenstrukturen |
| Dozent(en) | Malte Heins |
| Hörtermin | 3 |
| Häufigkeit | jährlich |
| Lehrform | Übung/Praktikum/Planspiel |
| Semesterwochenstunden | 2 |
| ECTS | 2.0 |
| Prüfungsform | Abnahme |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, studentische Arbeit am Rechner, Tutorien |

Lernziele

Die Studierenden ...

- wenden praktisch die Inhalte aus der Vorlesung an.
- vertiefen die Beherrschung der Programmiersprache Java und der objektorientierten Programmierung.
- erlangen die Fähigkeit zur Erstellung algorithmenorientierter Programme in Java.

Inhalt

Bearbeitung von Übungsaufgaben parallel zum Stoff der Vorlesung in Zweiergruppen mit Abnahme und Diskussion der Lösungen. Zusätzlich werden praxisrelevante Aspekte der Anwendungsentwicklung mit der Programmiersprache Java behandelt, die nicht Bestandteil der Vorlesung sind wie z. B. Dateieingabe und -ausgabe.

Literatur

- Unterlagen zur Übung im Web

I.1.15 Systemnahe Programmierung

B043 Systemnahe Programmierung

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B043 |
| Bezeichnung | Systemnahe Programmierung |
| Lehrveranstaltung(en) | B043a Systemnahe Programmierung B043b Übg. Systemnahe Programmierung |
| Verantwortliche(r) | M.Sc. Christian Uhlig |
| Zuordnung zum Curriculum | Computer Games Technology (Bachelor) IT-Ingenieurwesen (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Smart Technology (Bachelor) Technische Informatik (Bachelor) |
| Verwendbarkeit | Das Modul setzt auf den konzeptionellen Inhalten des Moduls „Programmstrukturen 1“ und der im Modul „Programmstrukturen 2“ erworbenen fortgeschrittenen Programmiererfahrung auf. Es kann mit anderen fortgeschrittenen Modulen zur Software-Technik kombiniert werden, insbesondere mit „Algorithmen und Datenstrukturen“, und schafft die notwendigen Voraussetzungen für Anschlussmodule (z.B. im Bereich der Computergrafik), die Kenntnisse in der Programmiersprache C erfordern. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Vorausgesetzt werden Grundkenntnisse in statisch getypten imperativen Programmiersprachen, die insbesondere charakteristische Datentypen und Kontrollstrukturen (Sequenz, Selektion, Iteration) umfassen. Diese Kenntnisse sollten mit gefestigter Programmierpraxis in einer entsprechenden Sprache verbunden sein. |
| Dauer | 1 |
| Lernziele | Nach Abschluss des Moduls kennen die Studierenden die Charakteristika und die wesentlichen Elemente der Programmiersprache C und können die Einsatzfelder der Sprache benennen. Sie wenden die erworbenen Kenntnisse zur Implementierung von Software in C praktisch an. Dabei |

setzen sie typische Sprachkonzepte wie Zeiger nebst Zeigerarithmetik, Arrays, dynamische Speicherverwaltung, bitweise Operationen und Funktionszeiger sowie den Präprozessor und die Standard-Bibliothek in sicherer und bezogen auf den Anwendungsfall möglichst portabler Form ein.

Weiterhin sind die Studierenden sensibilisiert für die inhärenten Probleme systemnaher Programmierung insbesondere in Hinblick auf unsichere Sprachkonzepte, schwache Typprüfungen, Portabilitätsprobleme durch plattformabhängiges Verhalten und Konsequenzen von Programmfehlern. Sie setzen diese Sensibilisierung in besondere Sorgfalt und Präzision bei der Programmierung in C um.

Sie können die Funktionsweise eines Rechners und die damit verbundenen Kosten bei der Ausführung von Programmen einer Hochsprache insbesondere bezüglich der Umsetzung von Datentypen, der Ausführung von Schleifen und Verzweigungen und der Implementierung von Unterprogrammaufrufen grob darstellen und im Zuge eigener Entwicklungstätigkeit einfließen lassen.

I.1.15.1 Systemnahe Programmierung

| | |
|---------------------------------|---|
| Lehrveranstaltung | Systemnahe Programmierung |
| Dozent(en) | Christian Uhlig |
| Hörtermin | 3 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 2 |
| ECTS | 2.0 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, Handout, interaktive Entwicklung und Diskussion von Modellen, Softwaredemonstration |

Lernziele

Die Studierenden ...

- formulieren Programme in der Programmiersprache C unter Berücksichtigung der Besonderheiten der Programmiersprache insbesondere in Hinblick auf undefiniertes Verhalten, Plattformabhängigkeiten und Unsicherheiten bestimmter Sprachkonstrukte (z.B. Zeigerarithmetik und fehlende Boundary Checks).
- erläutern in groben Zügen die Repräsentation von Daten und die Abläufe in einem Rechner bei der Ausführung von Anweisungen und Auswertung von Ausdrücken in einer höheren Programmiersprache, insbesondere im Rahmen von Unterprogrammaufrufen.

Inhalt

- Typische Elemente und Eigenschaften eines C-Programms
- Datentypen
 - Ganzzahl- und Aufzählungstypen
 - Fließkommatypen, Grundlagen von Fließkommazahlen
 - Strukturierte Typen
 - Vereinigungstypen
 - Zeigertypen
 - Arraytypen
- Funktionszeiger und ihre Anwendungsbereiche
- Konvertierungen
- Arrays und ihre Beziehung zu Zeigern
- Ausdrücke
 - Konstanten
 - Grundlegende Ausdrücke (Zuweisungen, Funktionsaufrufe, etc.)

- Arithmetische Ausdrücke
- Boolesche Ausdrücke, Vergleichsoperatoren, logische Operatoren
- Bitweise Operatoren
- Arbeit mit Zeigern und Zeigerarithmetik
- Vorrang und Assoziativität
- Anweisungen, insbesondere Verzweigungen und Schleifen
- Dynamische Speicherverwaltung
- Übersetzungsprozess und C-Präprozessor
- Funktionsaufrufe in Maschinen, Aufrufstapel
- Gefahren der Sprache C am Beispiel eines Buffer Overflows mit Manipulation der Rücksprungadresse

Literatur

- Uhlig, Christian: Systemnahe Programmierung, Vorlesungsunterlagen im Web: <http://www.fh-wedel.de/~uhl/sp.html>
- Harbison, Samuel; Steele, Guy L.: C - A Reference Manual, 5th edition, Prentice Hall, New Jersey, 2002
- Kernighan, Brian W.; Ritchie, Dennis M.: C Programming Language, Prentice Hall, New Jersey, 1998

I.1.15.2 Übg. Systemnahe Programmierung

| | |
|---------------------------------|--|
| Lehrveranstaltung | Übg. Systemnahe Programmierung |
| Dozent(en) | Malte Heins |
| Hörtermin | 3 |
| Häufigkeit | jährlich |
| Lehrform | Übung/Praktikum/Planspiel |
| Semesterwochenstunden | 2 |
| ECTS | 3.0 |
| Prüfungsform | Abnahme |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, studentische Arbeit am Rechner, Tutorien |

Lernziele

Die Studierenden ...

- wenden praktisch die Inhalte aus der Vorlesung an.
- beherrschen die Programmiersprache C und die maschinennahen Konzepte der Sprache.
- erlangen die Fähigkeit zur Erstellung maschinennaher Programme.
- erlangen das Verständnis über die Abläufe in einer Maschine bei der Ausführung von Sprachkonstrukten aus höheren Programmiersprachen, wie zum Beispiel die Laufzeitor-
ganisation bei Funktionsaufrufen.

Inhalt

Bearbeitung von Übungsaufgaben parallel zum Stoff der Vorlesung in Zweiergruppen mit Abnahme und Diskussion der Lösungen. Zusätzlich werden praxisrelevante Aspekte der Anwendungsentwicklung mit der Programmiersprache C behandelt, die nicht Bestandteil der Vorlesung sind.

Literatur

- Unterlagen zur Übung im Web
- siehe auch Vorlesung Systemnahe Programmierung

I.1.16 Einführung in Datenbanken

B052 Einführung in Datenbanken

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B052 |
| Bezeichnung | Einführung in Datenbanken |
| Lehrveranstaltung(en) | B052a Einführung in Datenbanken B052b Übg. Einführung in Datenbanken |
| Verantwortliche(r) | Dr. Michael Predeschly |
| Zuordnung zum Curriculum | Betriebswirtschaftslehre (Bachelor) Computer Games Technology (Bachelor) Data Science & Artificial Intelligence (Bachelor) E-Commerce (Bachelor) IT-Ingenieurwesen (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Smart Technology (Bachelor) Technische Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) Wirtschaftsingenieurwesen (Bachelor) |
| Verwendbarkeit | Das Modul komplementiert Einführungen in die Programmierung ("Einführung in die Programmierung", "Programmstrukturen 1") in allen Studiengängen. Es ist mit den fortgeschrittenen Modulen "Datenbanktheorie und -implementierung" (Bachelor) und "Konzepte der Datenbanktechnologie" (Master) kombinierbar. Das Modul sollte in allen Studiengängen verwendet werden, in denen Datenhaltung wesentlich ist. |
| Semesterwochenstunden | 3 |
| ECTS | 5.0 |
| Voraussetzungen | Vorausgesetzt werden Grundkenntnisse in Programmierung und die Fähigkeit, abstrakt zu denken. |
| Dauer | 1 |
| Lernziele | Nachdem Studierende die Veranstaltungen des Moduls besucht haben, haben sie die Fähigkeit, eine relationale Datenbank unter Nutzung von SQL abzufragen, einzurichten und die betrieb- |

liche Informationsverarbeitung mittels relationaler Datenbanksysteme unter Nutzung von SQL zu planen und durchzuführen. Zudem haben sie die Fähigkeit, selbständig einen Datenbankentwurfsprozess unter Verwendung des Entity-Relationship-Datenmodells und des relationalen Datenmodells durchzuführen.

I.1.16.1 Einführung in Datenbanken

| | |
|---------------------------------|--|
| Lehrveranstaltung | Einführung in Datenbanken |
| Dozent(en) | Michael Predeschly |
| Hörtermin | 3 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 2 |
| ECTS | 3.0 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, E-Learning, Online-Aufbereitung, Tafel |

Lernziele

Die Studierenden ...

- beherrschen die Grundlagen der relationalen Datenbanktechnologie;
- erlangen die Fähigkeit, selbstständig einen Datenbankentwurfsprozess zu planen, eine relationale Datenbank unter Nutzung von SQL einzurichten und die Informationsverarbeitung mittels relationaler Datenbanksysteme unter Nutzung von SQL durchzuführen;
- erlangen die Fähigkeit, mit einem Entwurfstool einen Datenbankentwurfsprozess durchzuführen und mittels SQL selbstständig Anfragen an ein Datenbanksystem zu stellen.

Inhalt

- Einführung in die Datenbanktechnologie
- Datenbanksprache SQL - Einführung
- Datenbank-Abfrage mit SQL
- Datenbanksprache SQL - Einrichten der Datenbank
- Das Entity-Relationship-Datenmodell
- Das Relationale Datenmodell
 - Relationenschemata und Datenabhängigkeiten
 - Relationale Datenbanken
 - Normalformen
- Datenbank - Lebenszyklus

Literatur

- Elmasri, Ramez; Navathe, Shamkant B.: Grundlagen von Datenbanksystemen. 3. Aufl. München: Pearson -Verlag, 2009.
- Meier, Andreas: Relationale Datenbanken Leitfaden für die Praxis. Berlin: Springer-Verlag, 2004.

- Vetter, Max: Aufbau betrieblicher Informationssysteme mittels konzeptioneller Datenmodellierung. 8. Aufl. Stuttgart: Vieweg-Teubner, 1998.
- Vossen, Gottfried: Datenmodelle, Datenbanksprachen und Datenbank-Management-Systeme. 5. Aufl. Oldenbourg: Oldenbourg-Wissenschaftsverlag, 2008.

I.1.16.2 Übg. Einführung in Datenbanken

| | |
|---------------------------------|---|
| Lehrveranstaltung | Übg. Einführung in Datenbanken |
| Dozent(en) | Marco Pawlowski |
| Hörtermin | 3 |
| Häufigkeit | jährlich |
| Lehrform | Übung/Praktikum/Planspiel |
| Semesterwochenstunden | 1 |
| ECTS | 2.0 |
| Prüfungsform | Abnahme |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Softwaredemonstration, studentische Arbeit am Rechner |

Lernziele

Die Studierenden ...

- besitzen die Fähigkeit, ein Datenbanksystem mit SQL zu befragen und in nicht-triviale textuelle Anfrageanforderungen in SQL zu überführen.
- haben grundlegende Kenntnisse über die Ausführung der von ihnen gestellten Anfragen.
- haben die Kompetenz, ein Datenbankentwurfswerkzeug grundlegend zu bedienen.

Inhalt

Vorlesungsbegleitende praktische Übungen in SQL und zum Datenbankentwurf

Literatur

Vorlesungsunterlagen

I.1.17 Rechnernetze

B037 Rechnernetze

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B037 |
| Bezeichnung | Rechnernetze |
| Lehrveranstaltung(en) | B037a Rechnernetze B037b Prakt. Rechnernetze |
| Verantwortliche(r) | Dipl.-Ing. (FH) Ilja Kaleck |
| Zuordnung zum Curriculum | Computer Games Technology (Bachelor) E-Commerce (Bachelor) IT-Ingenieurwesen (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Smart Technology (Bachelor) Technische Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) |
| Verwendbarkeit | Das Modul ist sinnvoll mit den Inhalten der Grundlagemodule "Informationstechnik" und "Programmstrukturen 1 und 2" zu kombinieren. |
| Semesterwochenstunden | 6 |
| ECTS | 5.0 |
| Voraussetzungen | Grundlegende Kenntnisse im Bereich der allgemeinen Informationstechnik; grundlegende Programmierkenntnisse in C, Objekt-Pascal (Delphi) oder Java erleichtern das Verständnis für Interprozesskommunikation im Rahmen gezeigter Beispielprogramme; Kenntnisse im Umgang mit aktuellen Desktop-Betriebssystemen (Windows, optional MacOS-X bzw. Linux) sind zur eigenständigen Durchführung praktischer Übungsanteile hilfreich. |
| Dauer | 1 |

Lernziele

Nach Beendigung dieses Moduls verfügen die Studierenden über fundiertes Wissen über den Aufbau, den Betrieb und die Arbeitsweise moderner Rechnernetze (Computer Networks); dieses sowohl in technischer Hinsicht als auch in Bezug auf den Ablauf der Kommunikation zwischen Prozessen in Unternehmensnetzen bzw. dem Internet. Die Studierenden beherrschen allgemeine

Grundlagen der Datenkommunikation und kennen den Aufbau eines universellen Kommunikationsmodells, erlernt am Beispiel des OSI-Referenzmodells.

Vertiefendes Wissen haben sie bezüglich des Aufbaus und die Kommunikation in der Internet-Architektur (IPv4, IPv6). Hierbei verfügen sie über Kenntnisse hinsichtlich der Eigenschaften der verfügbaren Transportprotokolle und haben das grundlegende Verständnis zur Realisierung einfacher Interprozesskommunikation.

Sie kennen die für den Betrieb eines IP-basierten Netzes essentiell notwendigen Anwendungsprotokolle und können dieses Wissen auch als Basis für die Gestaltung eigener Anwendungen sinnvoll nutzen. Ferner verfügen sie über Kenntnisse hinsichtlich der Theorie und den praktischen Einsatz von Verzeichnisdiensten zur Verwaltung größerer Netze.

Darüber hinaus haben sie ein hinreichendes Verständnis für den technischen Aufbau und den Betrieb moderner Unternehmensnetze. Hierzu gehören fundierte Kenntnisse über die Eigenschaften aktueller Netztechnologien im Bereich Lokaler Netze (LANs) als auch drahtloser Netze (WLANs).

Sie kennen auch die Arbeitsweise der dabei eingesetzten Koppellemente und deren Vermittlungsstrategien zum Aufbau größerer Netzstrukturen bzw. des Internets.

Durch den praktischen Anteil des Moduls verfügen die Studierenden über ein vertieftes und punktuell auch signifikant ergänztes Wissen zuvor behandelter Lehrinhalte, eigenständig erlernt am eigenen PC-System (Server) im zugehörigen Schulungslabor. Sie verfügen auch über ein praxisnahes Verständnis über den realen Datenfluss in Netzen und können so typische Fehlersituationen im Rahmen der Kommunikation zwischen Anwendungen analysieren und eigenständig beheben. Diese Fähigkeit bildet eine wesentliche Grundlage für eine effiziente Entwicklung verteilter Anwendungen im Rahmen komplexer Softwareprojekte.

I.1.17.1 Rechnernetze

| | |
|---------------------------------|--|
| Lehrveranstaltung | Rechnernetze |
| Dozent(en) | Ilja Kaleck |
| Hörtermin | 3 |
| Häufigkeit | jedes Semester |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 4 |
| ECTS | 3.0 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | E-Learning, Online-Aufbereitung, Softwaredemonstration |

Lernziele

Die Studierenden erlangen ...

- ein grundlegendes Verständnis für den Aufbau einer herstellerneutralen Kommunikationsarchitektur (OSI).
- Kenntnisse über den Aufbau und die Funktion des Internet-Architekturmodells.
 - Kenntnis über IPv4-Adress- und Netzstrukturen.
 - Verständnis über die Arbeitsweise essentieller Anwendungsprotokolle.
 - Fähigkeit zum Verständnis des Ablaufs einfacher Interprozesskommunikation, u.a. als Basis für die Realisierung komplexerer verteilter Anwendungen.
 - die Arbeitsweise spezifischer Maßnahmen gegen den IPv4-Adressmangel im IPv4 (NAT, Proxyserver-Dienste) kennen.
 - Wissen über die Eigenschaften des neuen Internet-Protokolls Version 6 (IPv6) und Änderungen an bestehenden Internet-Protokollen (u. a. DNS, ICMP).
- Verständnis über den technischen Aufbau und den Betrieb Lokaler Netze (LANs).
 - Verständnis hinsichtlich des generellen Ablaufs der IP-Kommunikation in LANs.
 - Wissen um die Eigenschaften aktueller Netztechnologien (Schwerpunkt: Ethernet-Technik).
 - Kenntnisse zum Aufbau und Betrieb drahtloser Netze (IEEE 802.11 WLANs).
- Wissen um den technischen Aufbau von Netzstrukturen bzw. des Internets.
 - Wissen um die Aufgabe Funktionsweise der klassischen von Koppelemente in Netzen.
 - elementares Wissen um die Arbeitsweise praxisrelevanter Routingverfahren für kleinere und größere Netze (u. a. einfaches IP-Routing; hierarchisches Routing).
- Grundkenntnisse über den Aufbau und die Funktionsweise von Verzeichnisdiensten.

Inhalt

- Allgemeine Grundlagen und Begriffe
 - Allgemeine Strukturen in der Datenkommunikation
 - Protokolle und Protokollabläufe
 - Netztopologien und Klassifizierung von Übertragungsnetzen
- Das ISO-OSI Referenzmodell
 - Prinzip der Schichtenbildung und Schichtenfunktionen im Überblick
 - Datenfluss im Modell
 - Aktuelle Koppellelemente zum Netzaufbau im Kontext der OSI-Modells
- Die Internet-Architektur
 - Historie, Architekturübersicht, Standardisierungen
 - IPv4-Adressstrukturen und Netzaufbau, Subnetting
 - UDP-/TCP-Kommunikation, Sockets bzw. Socket-Kommunikation
 - Betrachtung ausgewählter Anwendungsprotokolle (DNS, TELNET / SSH, SMTP, HTTP, ...)
 - Network Address Translation (NAT) und der Einsatz von Proxy-Servern
 - Einführung in das neue Internet Protocol Version 6 (IPv6)
 - * Adress- und Netzstruktur, Migrationshinweise
 - * Änderungen an höheren Protokollen in Bezug auf das IPv6
- Technik Lokaler Netze (LANs)
 - Ablauf der Kommunikation in IEEE 802 LANs (Layer-2, IP, inkl. DHCP)
 - Schwerpunktbetrachtung: Ethernet-Technik, Zugriffsverfahren und
 - Technische Umsetzungen (10Mbps / 100FE / 1GbE / 10GbE)
 - Überblick über andere LAN-Technologien
- Koppellelemente und Vermittlungstechniken
 - Repeater, Brücken- bzw. Layer-2 Switching-Technologie
 - Virtuelle LANs (VLANs), Class-of-Services im LAN
 - Router bzw. IP-Routing, Link-State und Distanzvektor-Verfahren,
 - Hierarchisches Routing und IP-Multicasting
 - Drahtlose Netze nach IEEE 802.11,
 - * Struktur, Aufbau, Übertragungskonzepte, Sicherheitsbetrachtungen
- Verzeichnisdiente
 - Einführung und grundlegendes Konzept des X.500

- Herstellerspezifische Lösungen (Active Directory)
- Lightweight Directory Access Protocol (LDAP)

Literatur

- TANNENBAUM, Andrew S.:
Computer Netzwerke.
5. Aufl. München: Pearson Education, 2012, ISBN 978-3-86894-137-1
- KUROSE, James F.; ROSS, Keith W.:
Computer Netzwerke. Der Top-Down Ansatz.
6. Aufl. : Pearson Education, 2014, ISBN 978-3-86894-237-8
- HALSALL, Fred:
Computer Networking and the Internet.
5. Aufl. München: Addison-Wesley, 2005, ISBN 978-0321263582
- RECH, Jörg:
Ethernet. Technologien und Protokolle für die Computervernetzung.
2. Aufl. Heidelberg: dPunkt-Verlag, 2007, ISBN 978-3-936931-40-2
- RECH, Jörg:
Wireless LANs. 802.11-WLAN-Technologie und praktische Umsetzung im Detail.
4. Aufl. Heidelberg: dPunkt-Verlag, 2012, ISBN 978-3-936931-75-4
- BADACH, Anatol; HOFFMANN, Erwin:
Technik der IP-Netze. Funktionsweise, Protokolle und Dienste.
2. Aufl. München: Hanser, 2007, ISBN 978-3446215016
- DAVIES, Joseph:
Understanding IPv6. Covers Windows 8 and Windows Server 2012.
3rd Edition: Microsoft Press, 2012, ISBN 978-0-7356-5914-8
- SCHÄFER, Günther:
Netzwerksicherheit. Algorithmische Grundlagen und Protokolle.
Heidelberg: dPunkt-Verlag, 2003, ISBN 3-89864-212-7
- SPERZEL Christian:
Netzwerksicherheit. Schützen Sie Ihr Netzwerk vor dem Zugriff anderer
Online-Videotrainig, Video2brain GmbH, 2014,
- BUEROSSE, Jörg:
Sichere E-Mails. Verschlüsselung und digitale Signatur unter Windows, Linux, OS X,
iOS und Android.
Online-Videotrainig, Video2brain GmbH, 2014
- FRISCH; HÖLZEL; LINTERMANN; SCHAÄFER:
Vernetzte IT-Systeme.
6. Aufl.:Bildungsverlag EINS, 2013, ISBN 978-3-8237-1141-4
- GRABA, Jan:
An Introduction to Network Programming with Java, Java 7 Compatible
3rd Edition: Springer-Verlag, 2013, ISBN 978-1-4471-5253-8

- CIUBOTARU, Bogdan ; MUNTEAN, Gabriel-Miro:
Advanced Network Programming - Principles and Techniques. Network Application Programming with Java.
Springer-Verlag, 2013, ISBN 978-1-4471-5291-0
- HAROLD, Elliotte Rusty:
Java Network Programming. Developing Networked Applications.
4th Edition, OReilly Media, 2013, ISBN 978-1-44935-767-2
- KLÜNTER, Dieter; LASER, Jochen:
LDAP verstehen, OpenLDAP einsetzen. Grundlagen und Praxiseinsatz.
2. Aufl. Heidelberg: dPunkt-Verlag, 2007, ISBN 978-3-89864-263-7

I.1.17.2 Prakt. Rechnernetze

| | |
|---------------------------------|--|
| Lehrveranstaltung | Prakt. Rechnernetze |
| Dozent(en) | Ilja Kaleck |
| Hörtermin | 3 |
| Häufigkeit | jedes Semester |
| Lehrform | Übung/Praktikum/Planspiel |
| Semesterwochenstunden | 2 |
| ECTS | 2.0 |
| Prüfungsform | Abnahme |
| Sprache | deutsch |
| Lehr- und Medienform(en) | E-Learning, Online-Aufbereitung, Softwaredemonstration, studentische Arbeit am Rechner |

Lernziele

Die Studierenden erlangen ...

- die Fähigkeit zum praktischen Umgang mit der Internet-Technologie am eigenen PC.
 - die Fähigkeit zum Anschluss von Systemen an ein Unternehmensnetz.
 - die Fähigkeit zur grundlegenden Konfiguration des Internet-Protokolls (IPv4, IPv6).
 - das Verständnis für Sicherheitsrichtlinien auf Multi-User Systemen (Windows, Linux).
 - die Fähigkeit zur Analyse und Behebung typischer Fehlersituationen im Rahmen der Kommunikation von Anwendungen und Systemen im Netz.
 - die Fähigkeit zur Konfiguration grundlegender Internet-Dienste (u. a. DNS, HTTP, FTP).
- das Verständnis für Lösungsansätze aktueller Techniken zur Unix-/Windows Integration in heterogenen Unternehmensnetzen (NFS, SAMBA, X-Windows).
- das Verständnis über aktuelle Konzepte zur Benutzer- und Rechteverwaltung in Netzen.
 - die Fähigkeit zur Benutzerverwaltung mittels eines Domänenkonzeptes (Windows).
 - die Fähigkeit zur Einrichtung von Verzeichnisdiensten (LDAP, Active Directory).
- die Grundkenntnisse zum praktischen Einsatz von Virtualisierungstechniken auf dem Desktop.
 - die Fähigkeit zur Einrichtung einfacher IP-Routingfunktionen auf einem System.
- das Verständnis über den praktischen Aufbau und Betrieb eines WLANs und dessen interne Kommunikationsabläufe (inkl. Sicherheitsbetrachtungen).
- die Fähigkeit zum Einsatz eines LAN-Analyzers zur Analyse von Kommunikationsabläufen zwischen Anwendungen sowie zur Fehleranalyse in LANs und WLANs.
- grundlegende Kenntnisse digitaler Sprachübertragung in Netzen mittels der Voice-over-IP (VoIP) Technik (Wahlthema).

- grundlegende Kenntnisse zu Streaming-Media Technik und den Real-Time Protokollen zur Übertragung multimedialer Inhalte in Netzen (Wahlthema).

Inhalt

Durchführung eines Laborpraktikums durchgängig individuell am eigenen PC-System unter Einsatz dedizierter Wechselfestplatten (Teilnehmer; Arbeitsgruppe)

- Einrichtung eines Server-Betriebssystems und Konfiguration der grundlegenden Kommunikationsprotokolle (IPv4, IPv6).
 - Nutzung typischer Internetdienstprogramme und Betrachtung der dabei verwendeten Protokolle.
- Einsatz von Techniken zur Unix/Windows-Integration (NFS, SAMBA, X-Windows, Unix mit Posix-ACLs)
- Nutzung einfacher Benutzer- und Rechteverwaltung im Netz (Domänenkonzept).
- Einsatz von Virtualisierungstechniken auf dem Desktop
 - Aufbau einer lokalen Netzinfrastruktur und Einrichtung des lokalen IP-Routings (inkl. NAT)
 - Grundlegende Firewall-Konfiguration
- Einrichten und Arbeiten mit aktuellen Verzeichnisdiensten
 - Aufbau einer eigenen Verzeichnisstruktur (Directory)
 - Formulierung von Suchanfragen an Verzeichnisdienste (Active Directory, LDAP-Server)
- Konfiguration grundlegender Internet-Serverdienste (DNS, FTP, HTTP, Proxy-Server, TELNET / SSH)
 - Nutzung der SSH Port-Forwarding Funktion
- Protokollanalyse und Fehlersuche im LAN mit einem LAN-Analyzer
 - Nutzung einer Remote-Probes zur verteilten LAN-Analyse im Netz.
 - Einfache LAN-Performance Messungen
- Konfiguration einer Arbeitsstation in einem Wireless-LAN (Adhoc und Infrastrukturnetz)
 - Analyse des drahtlosen Daten- und Kontrollverkehrs mit einem WLAN-Analyzer
- Einrichtung eines Voice-over-IP (VoIP) Clients (Wahlaufgabe)
 - Betrachtung dabei genutzter VoIP-Technologien und Übertragungsprotokolle
 - Einsatz eines LAN-Analyzers zur VoIP-Übertragungsanalyse
- Einführung in die Multi-Media Übertragung in Netzen (Wahlaufgabe)
 - Einrichtung eines eines aktuellen Streaming-Servers
 - Betrachtung der beteiligten Realtime-Übertragungsprotokolle
- Weitere Wahlthemen nach Aktualität.

Literatur

- RECH, Jörg:
Wireless LANs. 802.11-WLAN-Technologie und praktische Umsetzung im Detail.
4. Aufl. Heidelberg: dPunkt-Verlag, 2012, ISBN 978-3-936931-75-4
- BADACH, Anatol:
Voice-over-IP. Grundlagen, Protokolle, Anwendungen, Migration, Sicherheit.
4. Aufl. München: Hanser, 2009, ISBN 978-3-446-41772-4
- LIU/MATTHEW/PARZIALE/DAVIS/FORRESTER/BRITT:
TCP/IP Tutorial and Technical Overview (PDF). 8th. Ed. 2006: IBM-Redbook Serie.
<http://www.redbooks.ibm.com/redbooks/> Aktualisierungsdatum 29.06.2014
- GROUPER IEEE 802.11: Aktuelle Spezifikationen zu IEEE 802.11.
<http://standards.ieee.org/getieee802/802.11.html> Aktualisierungsdatum 29.06.2014
- IETF: Internet-Draft Dokumente und aktuelle RFCs.
<http://www.ietf.org/> - Aktualisierungsdatum 29.06.2014
- CISCO SYSTEMS: Internetworking Technology Handbook.
<http://www.cisco.com/c/en/us/td/docs/internetworking/technology/handbook/itodoc.html>
Aktualisierungsdatum 29.06.2014
- SPERZEL, Christian:
Netzwerksicherheit. Schützen Sie Ihr Netzwerk vor dem Zugriff anderer
Online-Videotrainig, Video2brain GmbH, 2014,
<https://www.video2brain.com/de/videotraining/netzwerksicherheit> - Aktualisierungsdatum 29.06.2014
- BUEROSSE, Jörg:
Sichere E-Mails. Verschlüsselung und digitale Signatur unter Windows, Linux, OS X, iOS und Android.
Online-Videotrainig, Video2brain GmbH, 2014,
<https://www.video2brain.com/de/videotraining/sichere-e-mails> - Aktualisierungsdatum 29.06.2014
- DIVERSE:
Schulungskurse zum Thema "Virtualisierung".
Online-Videotrainig, Video2brain GmbH, 2013,
<https://www.video2brain.com/de/search.htm?searchentry=Virtualisierung> - Aktualisierungsdatum 29.06.2014
- WOWZA MEDIA SYSTEMS:
Online Dokumentation zur "Wowza Streaming Engine"
<http://www.wowza.com/forums/content.php?188-documentation> - Aktualisierungsdatum 29.06.2014

I.1.18 Lineare Algebra

B045 Lineare Algebra

| | |
|---------------------------------|--|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B045 |
| Bezeichnung | Lineare Algebra |
| Lehrveranstaltung(en) | B045a Lineare Algebra |
| Verantwortliche(r) | Dr. Andreas Haase |
| Zuordnung zum Curriculum | Computer Games Technology (Bachelor) Data Science & Artificial Intelligence (Bachelor) IT-Ingenieurwesen (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Technische Informatik (Bachelor) |
| Verwendbarkeit | Das Modul "Lineare Algebra" baut auf den in der Veranstaltung "Grundlagen der Linearen Algebra" aus dem Modul "Deskriptive Statistik und Grundlagen der Linearen Algebra" erworbenen Kenntnissen und Fähigkeiten auf. Die im Modul "Lineare Algebra" erworbenen Kompetenzen stellen die Grundlage für zum Beispiel die weiterführenden Module "Grundlagen der Computergrafik", "Systemmodellierung" oder "Bildbearbeitung und -analyse" dar. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Das Modul setzt grundlegende Kenntnisse der linearen Algebra voraus, wie sie zum Beispiel im Modul "Deskriptive Statistik und Grundlagen der Linearen Algebra" vermittelt werden. |
| Dauer | 1 |

Lernziele

In diesem Modul werden weiterführende mathematische Kenntnisse aus dem Bereich der linearen Algebra, wie sie für ein quantitativ ausgerichtetes Studium unerlässlich sind, vermittelt. Der Schwerpunkt liegt dabei auf der Vektorraumtheorie und der analytischen Geometrie.

Die Lernenden sind in der Lage ausgewählte Problemstellungen aus den Bereichen Naturwissenschaft, Technik und Informatik, mittels der im Modul vermittelten mathematischen Methoden

zu modellieren und analysieren. Die Lernenden können für die vermittelten Inhalte praxisrelevante Anwendungsbeispiele benennen. Die Lernenden können eigenständig Lösungsmethoden für ausgewählte Problemstellungen auswählen, die Lösungsmethodik bis zum Ergebnis durchführen und die erhaltenen Ergebnisse kritisch bewerten

I.1.18.1 Lineare Algebra

| | |
|---------------------------------|------------------------------------|
| Lehrveranstaltung | Lineare Algebra |
| Dozent(en) | Andreas Haase |
| Hörtermin | 3 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, Handout, Tafel |

Lernziele

Nach dem erfolgreichen Besuch der Veranstaltung können die Studierenden ...

- die Determinante einer Matrix beliebiger Dimension berechnen und den Zusammenhang zur Lösungstheorie linearer Gleichungssysteme herstellen.
- die Vektorraumaxiome nennen und eine gegebene Menge mit Verknüpfungen darauf überprüfen ob diese ein Vektorraum (über \mathbb{R} oder \mathbb{C}) ist.
- Die Definition eines Unterraums nennen; Teilmengen von Vektorräumen darauf überprüfen ob diese Unterräume sind.
- das Konzept der linearen Abhängigkeit von Vektoren erklären; Teilmengen von Vektorräumen auf lineare Abhängigkeit überprüfen.
- die Definition einer Basis nennen. Teilmengen von Vektorräumen darauf überprüfen ob diese eine Basis sind.
- die Definition eines Skalarproduktes nennen; verschiedene lineare Abbildungen auf Vektorräumen darauf überprüfen ob diese ein Skalarprodukt sind.
- die Definition einer Norm nennen; den Zusammenhang zwischen Skalarprodukt und Norm nennen.
- Die Definition einer Orthonormalbasis nennen; eine Orthonormalbasis aus einer gegebenen Basis konstruieren (Gram-Schmidt-Verfahren).
- die Parameter und Koordinatendarstellung von Geraden und Ebenen formulieren; Lagebeziehungen zwischen linearen geometrischen Objekten berechnen; Lagebeziehungen zwischen linearen und einfachen nichtlinearen Geometrischen Objekten berechnen.
- die Definition einer linearen Abbildung nennen; lineare Abbildungen mittels Matrix-Vektor-Schreibweise ausdrücken. Eigenschaften gegebener linearer Abbildungen bestimmen.
- die Definition einer affinen Abbildung nennen; affine Abbildungen mittels Matrix-Vektor-Schreibweise ausdrücken. Eigenschaften gegebener affiner Abbildungen bestimmen.
- Koordinatentransformationen als affine Abbildung durchführen; die affine Abbildung einer Koordinatentransformation berechnen; aktive und passive Koordinatentransforma-

tionen unterscheiden.

- das charakteristische Polynom einer Matrix aufstellen; die Eigenwerte einer Matrix berechnen; die Eigenvektoren einer Matrix berechnen.
- eine Matrix diagonalisieren.
- bestimmte Funktionen einer Matrix berechnen.

Inhalt

- Wiederholung: Grundlagen der linearen Algebra
- Determinanten
 - der Entwicklungssatz von Laplace
 - lineare Gleichungssysteme
- Vektorräume
 - Definition, Beispiele und Eigenschaften
 - Unterräume
 - Lineare Abhängigkeit, Basis und Dimension
- Euklidische und unitäre Vektorräume
 - Skalarprodukt und Norm
 - Orthogonalität
 - Orthogonal- und Orthonormalbasen
- Analytische Geometrie
 - Darstellung von Geraden und Ebenen
 - Lagebeziehung zwischen linearen geometrischen Objekten
 - Einfache nichtlineare Objekte am Beispiel
- Abbildungen
 - Lineare Abbildungen
 - Affine Abbildungen
 - Koordinatentransformationen
- Eigenwerte und Eigenvektoren
 - Charakteristisches Polynom, Eigenwerte, Eigenvektoren
 - Diagonalisierung
 - Matrixfunktionen

Literatur

- GRAMLICH, Günter M.:
Lineare Algebra: Eine Einführung.
3. aktualisierte Aufl. München: Carl Hanser Verlag 2011
- FISCHER, Gerd:
Lernbuch Lineare Algebra und Analytische Geometrie.
1. Aufl. Wiesbaden: Vieweg + Teubner Verlag 2011
- FARIN, Gerald; HANSFORD, Dianne:
Lineare Algebra: Ein geometrischer Zugang,
Springer Verlag 2003
- FISCHER, Gerd:
Lineare Algebra: Eine Einführung für Studienanfänger.
18., aktualisierte Aufl. Wiesbaden: Springer Verlag 2013
- LIESEN, Jörg; MEHRMANN, Volker:
Lineare Algebra: Ein Lehrbuch über die Theorie mit Blick auf die Praxis.
1. Aufl. Wiesbaden: Vieweg + Teubner Verlag 2011
- ZIESCHANG, Heiner:
Lineare Algebra und Geometrie.
1. Aufl. Stuttgart, Teubner Verlag 1997

I.1.19 Einführung in die Betriebswirtschaft

B034 Einführung in die Betriebswirtschaft

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B034 |
| Bezeichnung | Einführung in die Betriebswirtschaft |
| Lehrveranstaltung(en) | B034a Einführung in die Betriebswirtschaft |
| Verantwortliche(r) | M.Sc. Fikret Koyuncu |
| Zuordnung zum Curriculum | Betriebswirtschaftslehre (Bachelor) Computer Games Technology (Bachelor) Data Science & Artificial Intelligence (Bachelor) E-Commerce (Bachelor) IT-Ingenieurwesen (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Smart Technology (Bachelor) Technische Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) Wirtschaftsingenieurwesen (Bachelor) |
| Verwendbarkeit | Das Modul "Einführung in die Betriebswirtschaft" ist ein Einführungsmodul. Die erworbenen Kompetenzen stellen wesentliche Grundlagen für eine Vielzahl weiterer Module dar, wie zum Beispiel "Operatives Produktionsmanagement", "Business Planning" oder "Controlling & Unternehmensführung". |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Keine |
| Dauer | 1 |

Lernziele

Nach erfolgreicher Teilnahme an diesem Modul können die Studierenden die Bedeutung von betriebswirtschaftlichen Denkweisen und Methoden für die moderne Unternehmensführung abschätzen. Sie kennen grundlegende Fragestellungen und Methoden zu deren Bearbeitung aus dem Bereich der Allgemeinen Betriebswirtschaftslehre. Die Studierenden sind befähigt, ausgewählte Aufgaben, wie sie sich in der Praxis des Unternehmens ergeben, unter Anwendung betriebswirtschaftlicher Methoden zu lösen.

Die Studierenden können wechselseitige Abhängigkeit zwischen den Aufgaben aus den Bereichen der Betriebswirtschaftslehre, den Ingenieurwissenschaften und der Informatik identifizieren und benennen.

I.1.19.1 Einführung in die Betriebswirtschaft

| | |
|---------------------------------|---|
| Lehrveranstaltung | Einführung in die Betriebswirtschaft |
| Dozent(en) | Fikret Koyuncu |
| Hörtermin | 3 |
| Häufigkeit | jedes Semester |
| Lehrform | Vorlesung mit integrierter Übung/Workshop/Assigm. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, Handout, Tafel |

Lernziele

Die Studierenden können ...

- das Erfahrungs- und Erkenntnisobjekt der Betriebswirtschaftslehre benennen,
- die Begriffe Wirtschaften und Ökonomisches Prinzip erklären sowie eine Break-Even-Analyse durchführen,
- Unternehmensziele aufzählen; die Aufgaben der Zielbildung erläutern sowie den Zielbildungsprozess wiedergeben,
- ausgewählte Kennzahlen ausrechnen,
- Ziele der Unternehmensführung erläutern, Führungsebenen voneinander abgrenzen, den Führungsprozess beschreiben sowie ausgewählte Führungsstile erläutern und -prinzipien erklären,
- Standortfaktoren identifizieren und Modelle zur Standortbewertung einsetzen,
- die Ziele der Materialwirtschaft wiedergeben und durch Anwendung von Methoden materialwirtschaftliche Analysen durchführen und Handlungsanweisungen ableiten,
- ausgewählte Erzeugnisstrukturdarstellungen für gegebene Problemstellungen erstellen und mit programmorientierten Verfahren die Materialbedarfsplanung durchführen,
- mit ausgewählten Verfahren die optimale Bestellmenge bestimmen,
- den Input, Throughput und Output von Produktionsprozessen beschreiben,
- das optimale Produktionsprogramm für ausgewählte Fälle ermitteln,
- ausgewählte Aufgaben der Produktionsprozessplanung ausführen,
- die Ziele des Marketings nennen, Methoden zur Ableitung der Marketing-Strategie beschreiben und anwenden sowie die Instrumente des Marketing-Mix erläutern,
- Investitionsarten voneinander abgrenzen; den Investitionsprozess beschreiben und die Aufgabe der Investitionskontrolle skizzieren sowie die Vorteilhaftigkeit einer Investition mittels Methoden beurteilen,
- die Ziele und Aufgaben der Finanzwirtschaft nennen; die Finanzierung aus Abschreibungen erläutern sowie den Financial-Leverage-Effekt an einem Beispiel demonstrieren,

- die Bedeutung informationstechnischer Systeme zur Bewältigung betriebswirtschaftlicher Aufgaben erläutern.

Inhalt

Die Studierenden erlernen Grundtatbestände der Betriebswirtschaftslehre, beginnend vom Erfahrungs- und Erkenntnisobjekt dieser wissenschaftlichen Disziplin, über zu fällende konstitutive Entscheidungen, bis hin zu den diversen betriebswirtschaftlichen Funktionen innerhalb eines Betriebes.

Letztere stehen im Mittelpunkt der Veranstaltung. Die theoretischen Inhalte werden durch Praxisbeispiele untersetzt.

Durch zahlreiche Übungen wird das Verständnis für die betriebswirtschaftlichen Prozesse und deren Zusammenhänge gefestigt sowie das eigenständige Arbeiten gefördert.

Inhalte der Veranstaltung sind im Einzelnen:

- Betriebswirtschaftslehre als wissenschaftliche Disziplin
- Grundlagen der Betriebswirtschaftslehre
- Standortwahl
- Unternehmensführung
- Materialwirtschaft
- Produktionswirtschaft
- Marketing & Absatz
- Investition & Finanzierung
- Umfangreiche Übungen zu verschiedenen Vorlesungsteilen

Literatur

- BECKER, Hans Paul: Investition und Finanzierung. 7. akt. Aufl. Wiesbaden: Gabler, 2016
- BERNECKER, Michael: Grundlagen der Betriebswirtschaftslehre. 4. Aufl. Köln: Johanna, 2011.
- BLOHM, Hans; LÜDER, Klaus; SCHÄFER, Christina: Investition. 10. akt. Aufl. München: Vahlen, 2012
- DÄUMLER, Klaus-Dieter; GRABE, Jürgen: Grundlagen der Investitions- und Wirtschaftlichkeitsrechnung. 12. vollst. überarbeitete Aufl. Berlin; Herne: Neue Wirtschafts-Briefe, 2007
- JUNG, Hans: Allgemeine Betriebswirtschaftslehre. akt. 13. Aufl. München: Oldenbourg, 2016
- SCHIERENBECK, Henner; WÖHLE, Claudia: Grundzüge der Betriebswirtschaftslehre. 18. überarb. Aufl. München: Oldenbourg, 2012
- SPECHT, Olaf; SCHMITT, Ulrich: Betriebswirtschaftslehre für Ingenieure + Informatiker. 5. Aufl. München; Wien: Oldenbourg, 2000

- THOMMEN, Jean-Paul; ACHLEITNER, Ann-Kristin: Allgemeine Betriebswirtschaftslehre. 7. vollst. überarb. Aufl. Wiesbaden: Gabler, 2012
- VAHS, Dietmar; SCHÄFER-KUNZ, Jan: Einführung in die Betriebswirtschaftslehre. 7. überarb. Aufl. Stuttgart: Schäffer-Poeschel, 2015.
- WEBER, Wolfgang; KABST, Rüdiger: Einführung in die Betriebswirtschaftslehre. 9. akt. u. überarb. Aufl. Wiesbaden: Gabler, 2014
- WÖHE, Günter; DÖRING, Ulrich: Einführung in die Allgemeine Betriebswirtschaftslehre. 26. überarbeitete und aktualisierte Aufl. München: Vahlen, 2016

I.1.20 Fortgeschrittene Objektorientierte Programmierung

B057 Fortgeschrittene Objektorientierte Programmierung

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B057 |
| Bezeichnung | Fortgeschrittene Objektorientierte Programmierung |
| Lehrveranstaltung(en) | B057a Fortgeschrittene Objektorientierte Programmierung B057b Übg. Fortgeschrittene Objektorientierte Programmierung |
| Verantwortliche(r) | M.Sc. Christian Uhlig |
| Zuordnung zum Curriculum | Computer Games Technology (Bachelor) Data Science & Artificial Intelligence (Bachelor) IT-Ingenieurwesen (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Smart Technology (Bachelor) Technische Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) |
| Verwendbarkeit | Das Modul setzt unmittelbar auf den Inhalten des Moduls „Programmstrukturen 2“ auf und eignet sich damit als Weiterqualifikation im Anschluss an „Programmstrukturen 2“ und das „Programmierpraktikum“. Es kann ergänzend insbesondere mit dem Modul „Software-Design“ kombiniert werden. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Vorausgesetzt werden gefestigte theoretische und praktische Kenntnisse in objektorientierter Programmierung im Allgemeinen und in der Programmiersprache Java im Speziellen. Dies sollte auch mindestens Einstiegskenntnisse zu generischen Typen (Java Generics) umfassen. |
| Dauer | 1 |

Lernziele

Nach Abschluss des Moduls beherrschen die Studierenden den methodisch fundierten praktischen Umgang mit fortgeschrittenen objektorientierten Sprachkonzepten am Beispiel von Java. Sie sind in der Lage, vorgegebene generische Typen zu nutzen und eigene generische Typen zu deklarieren sowie in problemadäquater Weise Funktionswerte und Prinzipien funktionaler Programmierung in objekt-orientierten Sprachen anzuwenden. Die Teilnehmer erkennen Nutzen und Probleme nebenläufiger Programmierung und werden in die Lage versetzt, Threads und Konzepte zur Thread-Synchronisation anzuwenden. Ergänzend erlernen die Studierenden die Grundzüge der Programmierung mit Reflexion und können die Java Reflection API in ihren fundamentalen Einrichtungen nebst Java-Annotationstypen praktisch anwenden.

I.1.20.1 Fortgeschrittene Objektorientierte Programmierung

| | |
|---------------------------------|---|
| Lehrveranstaltung | Fortgeschrittene Objektorientierte Programmierung |
| Dozent(en) | Christian Uhlig |
| Hörtermin | 4 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 2 |
| ECTS | 2.0 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, Handout, interaktive Entwicklung und Diskussion von Modellen, Softwaredemonstration |

Lernziele

Die Studierenden ...

- nennen und erläutern die methodischen Grundlagen von objektorientierten Programmiersprachen am Beispiel von Java.
- wenden fortgeschrittene Konzepte objektorientierter Programmiersprachen wie parametrische Polymorphie, Mehrfachvererbung und Funktionswerte zur Konstruktion wiederverwendbarer Softwarekomponenten an.
- nennen und erläutern die Grundlagen generischer Datentypen und ihre Korrespondenz mit Konzepten objektorientierter Sprachen (Schnittstellen, abstrakte Klassen, konkrete Klassen, Polymorphie, Verträge).
- entwerfen und implementieren generische Datentypen.
- definieren Funktionswerte per anonymer Klassen, Lambda-Ausdrücke und Methodenreferenzen.
- wenden Prinzipien funktionaler Programmierung in objektorientierten Sprachen am Beispiel von Java Streams an.
- nennen und erläutern Motivation, Grundlagen und Herausforderungen nebenläufiger Programmierung.
- wenden Primitiven nebenläufiger Programmierung in Java an (Erzeugen von Threads, Thread-Kommunikation / Synchronisation, usw.).
- nennen und erläutern die Vorzüge funktionaler Programmierung für die parallele Auswertung von Berechnungen am Beispiel von Java Streams.
- definieren Stream-Pipelines unter Berücksichtigung der Anforderungen und Konsequenzen einer nebenläufigen Auswertungsstrategie.
- nennen und erläutern die Grundzüge von Reflection in Programmiersprachen am Beispiel von Java.
- wenden die Java Reflection API i.V.m. Annotationstypen an.

Inhalt

- Generische Typen / Java Generics
- Verschachtelte Typen
- Funktionswerte in OO-Sprachen (Funktionale Interfaces, Lambda-Ausdrücke, Methodenreferenzen)
- Funktionale Programmierung mit Java Streams
- Nebenläufigkeit (Threads, Racing Conditions, Synchronisation, Waitsets, volatile Variablen, Java Memory Model, nebenläufige Auswertung von Streams, Executors, Futures, Fork-Join-Tasks, Thread-sichere Collections)
- Reflection, Annotationstypen

Literatur

- Uhlig, Christian: Fortgeschrittene Objektorientierte Programmierung, Vorlesungsunterlagen im Web: <http://www.fh-wedel.de/~uhl/foop.html>
- Gosling, James; Joy, Bill; Steele, Guy; Bracha, Gilad; Buckley, Alex: The Java Language Specification, Java SE 8 Edition, 2015
- Meyer, Bertrand: Objektorientierte Softwareentwicklung, Hanser, München, 1990
- Bloch, Joshua: Effective Java: A Programming Language Guide, 3rd Edition, Addison-Wesley, 2017
- Urma, Raoul-Gabriel; Fusco, Mario; Mycroft, Alan: Java 8 in Action: Lambdas, Streams, and Functional-Style Programming, Manning Publications, 2014
- Ullenboom, Christian: Java ist auch eine Insel, 12. Auflage, Galileo Press GmbH, 2016
- Louden, Kenneth C.: Programming Languages: Principles and Practice, 2nd Edition, Thomson Learning, 2002

I.1.20.2 Übg. Fortgeschrittene Objektorientierte Programmierung

| | |
|---------------------------------|--|
| Lehrveranstaltung | Übg. Fortgeschrittene Objektorientierte Programmierung |
| Dozent(en) | Malte Heins |
| Hörtermin | 4 |
| Häufigkeit | jährlich |
| Lehrform | Übung/Praktikum/Planspiel |
| Semesterwochenstunden | 2 |
| ECTS | 3.0 |
| Prüfungsform | Abnahme |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, studentische Arbeit am Rechner, Tutorien |

Lernziele

Die Studierenden ...

- wenden praktisch die Inhalte aus der Vorlesung an.

Inhalt

Bearbeitung von Übungsaufgaben parallel zum Stoff der Vorlesung in Zweiergruppen mit Abnahme und Diskussion der Lösungen. Zusätzlich werden praxisrelevante Aspekte der fortgeschrittenen OOP behandelt, die nicht Bestandteil der Vorlesung sind.

Literatur

- Unterlagen zur Übung im Web
- siehe auch Vorlesung Fortgeschrittene Objektorientierte Programmierung

I.1.21 Datenbanktheorie und -implementierung

B117 Datenbanktheorie und -implementierung

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B117 |
| Bezeichnung | Datenbanktheorie und -implementierung |
| Lehrveranstaltung(en) | B117a Datenbanktheorie und -implementierung |
| Verantwortliche(r) | Dr. Michael Predeschly |
| Zuordnung zum Curriculum | E-Commerce (Bachelor) Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) |
| Verwendbarkeit | Das Modul ist sinnvoll mit dem grundlegenden Module "Einführung in Datenbanken" kombinierbar. In einem konsekutiven Studiengang steht es im Zentrum eines Datenbank-Curriculums und lässt sich gut durch das Modul "Konzepte der Datenbanktechnologie" (Master) ergänzen. Das Modul sollte in allen Studiengängen verwendet werden, in denen Datenhaltung ein Studienschwerpunkt ist. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Es werden grundlegende Kenntnisse in der Benutzung von Datenbanksystemen und Kenntnisse in SQL vorausgesetzt. |
| Dauer | 1 |

Lernziele

Nach Besuch der Veranstaltungen dieses Moduls besitzen die Studierenden eingehende Kenntnisse der für die Implementierung von Datenbanksystemen wichtigen Architekturprinzipien, ihrer Datenstrukturen und Algorithmen. Sie besitzen die Fähigkeit Datenbanksysteme in Hinblick auf ihren Aufbau und der internen Funktionsweise zu bewerten und zu beurteilen. Sie sind in der Lage selbst Modifikationen an Datenbanksystem-Implementierungen zu planen und durchzuführen. Die Studierenden besitzen die Fähigkeit, die Arbeitsweise von Datenbanksystemen zu optimieren und auch selbst Architekturen für die Datenhaltung zu entwerfen. Sie besitzen grundlegende Kenntnisse der Administration von Datenbanken.

I.1.21.1 Datenbanktheorie und -implementierung

| | |
|---------------------------------|--|
| Lehrveranstaltung | Datenbanktheorie und -implementierung |
| Dozent(en) | Michael Predeschly |
| Hörtermin | 4 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung mit integrierter Übung/Workshop/Assigm. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, E-Learning, Online-Aufbereitung, Tafel |

Lernziele

Die Studierenden besitzen ...

- Kenntnisse der relationalen Datenbanktheorie und ihrer wesentlichen Fragestellungen;
- detaillierte Kenntnisse der für die Implementierung von Datenbanksystemen wichtigen Architekturprinzipien, Datenstrukturen und Algorithmen und damit des Aufbau und der interne Arbeit eines großen komplexen Softwaresystems;
- die Fähigkeit, die Arbeitsweise von Datenbanksystemen zu optimieren bzw. selbst Architekturen für große komplexe Softwaresysteme zu entwerfen;
- Fähigkeit eines Datenbankadministrators für Datenbanksysteme.

Inhalt

- Relationale Datenbanksysteme
 - Das 3 - Ebenen - Architekturkonzept
 - Transaktionskonzept
 - Relationale Operationen und Relationenalgebra
 - Komponenten eines DBMSs und Betrieb eines DBSs
- Aufbau eines Datenbankverwaltungssystems (DBMS)
 - Systementwurf und Schichtenmodell
 - Architektur eines DBMS
 - Komponenten eines DBMS
- Speichersystem
 - Speicherverwaltung
 - Systempufferverwaltung
- Speicherungsstrukturen und Zugriffspfade

- Freispeicherverwaltung
 - Abbildung von Datensätzen in Seiten
 - Zugriffspfade
- Satzorientierte Datenbankschnittstelle
- Sicherung der Integrität in Datenbanken
 - Semantische Integrität
 - Operationale Integrität
 - Fehlerbehandlung - Recovery
- Grundlagen NoSQL-Datenbanken

Literatur

- Elmasri, Ramez; Navathe, Shamkant B.:
Grundlagen von Datenbanksystemen.
3. Auflage. München: Pearson -Verlag, 2009.
- Saake, Gunter; Sattler, Kai-Uwe; Heuer, Andreas:
Datenbanken Implementierungstechniken.
3. Auflage. mitp-Verlag, 2011
- Knuth, E. Donald: The Art of Computer Programming
Addison Wesley 2011

I.1.22 Software-Design

B058 Software-Design

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B058 |
| Bezeichnung | Software-Design |
| Lehrveranstaltung(en) | B058a Software-Design |
| Verantwortliche(r) | M.Sc. Christian Uhlig |
| Zuordnung zum Curriculum | Computer Games Technology (Bachelor) E-Commerce (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Smart Technology (Bachelor) Technische Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) |
| Verwendbarkeit | Das Modul setzt unmittelbar auf den Inhalten des Moduls „Programmstrukturen 2“ auf und eignet sich damit als Weiterqualifikation im Anschluss an „Programmstrukturen 2“. Es kann mit fortgeschrittenen Modulen zur Software-Technik kombiniert werden, insbesondere mit dem „Programmierpraktikum“, „Software-Projekt“ und „Fortgeschrittene Objektorientierte Programmierung“. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Vorausgesetzt werden Kenntnisse in imperativen und speziell objektorientierten Programmiersprachen sowie konkret in der Programmiersprache Java. Die Inhalte der Module „Algorithmen und Datenstrukturen“ sowie „Fortgeschrittene Objektorientierte Programmierung“ können gewinnbringend eingeordnet werden, sind aber nicht erforderlich. |
| Dauer | 1 |

Lernziele

Nach Abschluss des Moduls kennen die Studierenden den Gegenstand und die Bedeutung des Softwareentwurfs im Allgemeinen und des objektorientierten Entwurfs im Speziellen und setzen typische Modellierungskonzepte wie UML für den Entwurf ein. Hierbei beachten sie u.a. Kon-

sistenz, Vollständigkeit, Erweiterbarkeit und Performance des modellierten Softwaresystems in Hinblick auf das entsprechende Fachkonzept.

Die Teilnehmer beherrschen den Einsatz gängiger objektorientierter Entwurfsmuster, erläutern ihre Vor- und Nachteile sowie Alternativen und wählen für gegebene Anwendungsfälle geeignete Entwurfsmuster aus. Insbesondere können sie in diesem Zuge Vererbung und Komposition als Konzepte im objektorientierten Entwurf einsetzen und in ihren Eigenschaften diskutieren.

Ferner leiten die Teilnehmer am Beispiel von Java das Gerüst eines Softwaresystems aus Entwurfsmodellen ab und implementieren die relevanten Operationen entsprechend Entwurf und Fachkonzept.

I.1.22.1 Software-Design

| | |
|---------------------------------|---|
| Lehrveranstaltung | Software-Design |
| Dozent(en) | Christian Uhlig |
| Hörtermin | 4 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, Handout, interaktive Entwicklung und Diskussion von Modellen, Softwaredemonstration |

Lernziele

Die Studierenden ...

- erkennen und erläutern die Einordnung des Entwurfs in den Softwareentwicklungsprozess.
- erkennen und erläutern die Bedeutung der Modellbildung im Softwaredesign.
- differenzieren informelle und formale Entwurfskonzepte.
- erkennen und erläutern die Bedeutung von Verträgen bei Entwurf und Implementierung abstrakter Datentypen.
- erkennen und erläutern das Entwurfparadigma Design by Contract am Beispiel der Programmiersprachen Eiffel und Java.
- differenzieren Vererbung und Komposition als zentrale Konzepte des objektorientierten Entwurfs und wählen problemadäquat aus.
- wenden informelle Notationen und Methoden (OMT, UML, ERD, ...) zur Modellierung eines Softwaresystems an.
- wenden formale Notationen (z.B. Haskell) zur Definition der Datenstrukturen und der Schnittstellen eines Softwaresystems an.
- differenzieren Entwurfsmuster auf Grundlage von Struktur, Motivation und Zielsetzung.
- wenden typische objektorientierte Entwurfsmuster zur Lösung von softwaretechnischen Problemstellungen an.
- nennen und erläutern sprachabhängige und -unabhängige Implementierungsaspekte bei der Anwendung von Entwurfsmustern.
- erkennen und erläutern die Anwendbarkeit und Kombinierbarkeit einzelner Entwurfsmuster.
- differenzieren Flexibilität und Effizienz bei der problembezogenen Auswahl und Anwendung von Entwurfsmustern.

Inhalt

- Einordnung und Bedeutung des Entwurfs im Softwareentwicklungsprozess
- Methoden, Techniken und Werkzeuge im Software-Entwurf
 - informelle Methoden/Notationen: OMT, UML, ERD
 - formale Methoden/Notationen: abstrakte Syntax / Haskell
- Verträge zwischen Softwarekomponenten, Design by Contract
- Objektorientierte Entwurfsmuster
 - Erzeugungsmuster
 - Strukturmuster
 - Verhaltensmuster
- Fallstudien

Literatur

- Uhlig, Christian: Software-Design, Vorlesungsunterlagen im Web: <http://www.fh-wedel.de/~uhl/swd.html>
- Balzert, Helmut: Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb, 3. Auflage, Spektrum, 2011
- Balzert, Helmut; Balzert, Heide; Koschke, Rainer; Lämmel, Uwe; Liggesmeyer, Peter; Quante, Jochen: Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering, 3. Auflage, Spektrum, 2009
- Gamma, Erich; Helm, Richard; Johnson, Ralph; Vlissides, John: Design Patterns: Entwurfsmuster als Elemente wiederverwendbarer objektorientierter Software, mitp, 2014
- Freeman, Eric; Robson, Elisabeth; Bates, Bert; Sierra, Kathy: Head First Design Patterns, OReilly, 2014
- Rumbaugh, James; Blaha, Michael; Premerlani, William; Eddy, Frederick; Lorensen, William: Objektorientiertes Modellieren und Entwerfen, Hanser, 1994
- Fowler, Martin: UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3. revidierte Auflage, Addison-Wesley, 2003
- Hutton, Graham: Programming in Haskell, 2. Auflage, Cambridge University Press, 2016

I.1.23 Web-Anwendungen

B059 Web-Anwendungen

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B059 |
| Bezeichnung | Web-Anwendungen |
| Lehrveranstaltung(en) | B059a Web-Anwendungen B059b Übg. Web-Anwendungen |
| Verantwortliche(r) | Dr. Michael Predeschly |
| Zuordnung zum Curriculum | Computer Games Technology (Bachelor) Data Science & Artificial Intelligence (Bachelor) E-Commerce (Bachelor) IT-Ingenieurwesen (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Smart Technology (Bachelor) Wirtschaftsinformatik (Bachelor) |
| Verwendbarkeit | Das Modul baut auf den Kompetenzen auf, die durch Module des Themenbereichs Programmierung in Informatik-Studiengängen, insbesondere "Programmstrukturen 1", "Programmstrukturen 2" und "Algorithmen und Datenstrukturen", vermittelt werden. Es schafft die Voraussetzungen für Module im fortgeschrittenen Studienverlauf, in denen Kenntnisse zur Realisierung von Web-Anwendungen benötigt werden. Dies kann beispielsweise in den Modulen "Software-Projekt", "E-Commerce Grundlagen" und der Bachelor-Thesis der Fall sein. |
| Semesterwochenstunden | 5 |
| ECTS | 5.0 |
| Voraussetzungen | Grundkenntnisse der imperativen Programmierung und der Konzepte imperativer Programmiersprachen und objektorientierter Basiskonzepte. |
| Dauer | 1 |
| Lernziele | Die Studierenden kennen die speziellen technischen Randbedingungen und Besonderheiten der Entwicklung von Web-Anwendungen im Vergleich zu lokal laufenden Applikationen. Sie kennen |

die wichtigen Konzepte zur Realisierung von Web-Anwendungen und die Sprachen, die bei der Erstellung und im Umfeld des Einsatzes von Web-Anwendungen zum Einsatz kommen.

Die Studierenden kennen die Möglichkeiten der Cascading Style Sheets und wesentliche Bestandteile der Programmiersprachen Javascript zur Realisierung von Web-Anwendungen. Sie kennen die Möglichkeiten des Einsatzes von Frameworks zur Unterstützung der Entwicklung und können den Nutzen solcher Frameworks einschätzen. Sie sind in der Lage, ausgewählte Frameworks zur Realisierung von Web-Anwendungen zu nutzen.

Sie kennen die wesentlichen Erweiterungen von HTML 5 im Vergleich zu früheren HTML-Versionen und können diese auszugsweise zur Realisierung von Webseiten einsetzen.

Sie können auf Basis dieser Kenntnisse eigenständig Web-Seiten realisieren, die einfache Formen der Dynamik sowohl client-seitig als auch server-seitig enthalten und die kennengelernten Konzepte integrativ nutzen.

I.1.23.1 Web-Anwendungen

| | |
|---------------------------------|--|
| Lehrveranstaltung | Web-Anwendungen |
| Dozent(en) | Michael Predeschly |
| Hörtermin | 4 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 3 |
| ECTS | 3.0 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, E-Learning, Gastreferenten, Online-Aufbereitung, Tafel |

Lernziele

Die Studierenden ...

- führen die technischen Randbedingungen des Internet auf und benennen ihre Auswirkungen.
- beschreiben die konzeptionellen Aspekte von Stylesheets und der zentralen Möglichkeiten zur Festlegung der Darstellung in den Cascading Stylesheets und nutzen diese zur Erzeugung angestrebter Darstellungsweisen.
- können responsive Web-Layouts erstellen
- kennen wichtige Konzepte, Sprachen, Frameworks und Architekturen zur Realisierung dynamischer Webseiten auf, wählen zwischen diesen problembezogen aus und nutzen sie zur Erstellung dynamischer Webseiten.
- geben die zusätzliche Konzepte und Sprachelemente von HTML 5 an und entwerfen damit Webseiten.
- nutzen die theoretisch vermittelten Inhalte zur eigenständigen Realisierung von Webanwendungen begrenzter Komplexität.

Inhalt

- Basiskonzepte des WWW
 - Klassische Auszeichnungsmöglichkeiten in HTML
 - HTML-Formulare und ihre Möglichkeiten
 - Style Sheets
 - CSS-Animationen
 - Templating
 - Responsive Design
- Dynamik in Web-Seiten mit Javascript
 - Client-seitige Dynamik

– Server-seitige Dynamik

- Asynchronous Javascript

Literatur

- WOLF, Jürgen: HTML5 und CSS - Das umfassende Handbuch 2019
- ACKERMANN, Philipp: JavaScript - Das umfassende Handbuch 2019
- LABORENZ, Kai: CSS: Das umfassende Handbuch. Galileo Computing, 2011
- GASSTON, Peter: Moderne Webentwicklung: Geräteunabhängige Entwicklung - Techniken und Trends in HTML5, CSS3 und JavaScript, dpunkt.verlag, 2014
- WORLD WIDE WEB CONSORTIUM: HTML 5. <http://www.w3.org/TR/2014/WD-html5-20140617/>

I.1.23.2 Übg. Web-Anwendungen

| | |
|---------------------------------|---|
| Lehrveranstaltung | Übg. Web-Anwendungen |
| Dozent(en) | Marco Pawlowski |
| Hörtermin | 4 |
| Häufigkeit | jährlich |
| Lehrform | Übung/Praktikum/Planspiel |
| Semesterwochenstunden | 2 |
| ECTS | 2.0 |
| Prüfungsform | Abnahme |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Softwaredemonstration, studentische Arbeit am Rechner |

Lernziele

Die Studierenden ...

- sind in der Lage, die in der Vorlesung vermittelten theoretischen Hintergründe selbst praktisch anzuwenden.
- haben umfangreiche Kenntnisse und praktische Erfahrungen zu den Themen HTML, CSS, serverseitiger Dynamik, clientseitiger Dynamik mit JavaScript und AJAX, Einsatz JSON zum Austausch von Daten zwischen Client und Server, Einsatz von Cookies und Sessions zum temporären Speichern von Daten.
- steigern ihre Teamfähigkeit durch intensive Arbeit in Zweiertteams und Kommunikation über auftretende Probleme in der ganzen Gruppe.

Inhalt

Bearbeitung von Übungsaufgaben, die sich am Stoff der Vorlesung orientieren, in Zweiergruppen mit Abnahme der Lösungen. Erstellt wird eine im Verlaufe der einzelnen Übungseinheiten komplexer werdende Web-Anwendung, wobei die einzelnen Schritte aufeinander aufbauen, so dass am Ende eine komplexe Web-Anwendung entsteht, die einen Großteil der in der Vorlesung erlernten Techniken und Konzepte nutzt.

Literatur

I.1.24 UNIX und Shell-Programmierung

B044 UNIX und Shell-Programmierung

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B044 |
| Bezeichnung | UNIX und Shell-Programmierung |
| Lehrveranstaltung(en) | B044a UNIX & Shell Programmierung B044a Übg. UNIX & Shell-Programmierung |
| Verantwortliche(r) | M.Sc. Helga Karafiat |
| Zuordnung zum Curriculum | Computer Games Technology (Bachelor) Data Science & Artificial Intelligence (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Technische Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) |
| Verwendbarkeit | Das Modul setzt auf den konzeptionellen Inhalten des Moduls "Programmstrukturen 1" und auf der dort erworbenen Programmiererfahrung auf. Es kann mit anderen fortgeschrittenen Modulen zur Software-Technik kombiniert werden, insbesondere mit "Systemnaher Programmierung" und den Themengebieten Betriebssysteme, Compilerbau, Skriptsprachen und Webentwicklung. Zudem schafft es Grundlagen und Verständnis für die Verarbeitung und Analyse großer Datenmengen (Data Science / Big Data). |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Vorausgesetzt werden Kenntnisse der Grundkonzepte imperativer Programmiersprachen hinsichtlich der Umsetzung algorithmischer Grundstrukturen und die Fähigkeit zur Erstellung von vollständigen Programmen begrenzter Komplexität. |
| Dauer | 1 |
| Lernziele | Nach Abschluss des Moduls kennen die Studierenden die grundlegende Funktionsweise von Unix-Systemen und sind mit der Verwendung von Skriptsprachen und Tools in diesem Umfeld vertraut. |

Sie kennen die Vor- und Nachteile von Skriptsprachen bei der Software-Entwicklung und haben ein Gefühl für Automation von wiederkehrenden Aufgaben und für die Verarbeitung von großen Datenmengen entwickelt. Ferner kennen sie die regulären Ausdrücke zur Verarbeitung von Texten und deren Mächtigkeit und Grenzen. Sie können reguläre Ausdrücke sicher auf Problemstellungen anwenden. Sie haben zudem ein Verständnis für Datenströme entwickelt und die einfache und elegante Art der Kombinierbarkeit von Programmen, insbesondere an Hand von Filtern und Pipes, verinnerlicht.

Sie kennen sowohl die Flexibilität als auch die Fehleranfälligkeit von dynamischen Sprachen und haben eine Vorstellung davon, wann und wie sich die Produktivität beim Arbeiten mit Skriptsprachen im Vergleich zu kompilierten Sprachen verändert.

Die Studierenden finden sich durch den Umgang mit Unix-Systemen nun auch in einem Umfeld zurecht, in dem ihnen lediglich eine textbasierte Konsole zur Interaktion mit einem System zur Verfügung steht.

I.1.24.1 UNIX & Shell Programmierung

| | |
|---------------------------------|---|
| Lehrveranstaltung | UNIX & Shell Programmierung |
| Dozent(en) | Helga Karafiat |
| Hörtermin | 4 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung mit integrierter Übung/Workshop/Assigm. |
| Semesterwochenstunden | 2 |
| ECTS | 2.0 |
| Prüfungsform | Abnahme |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, Handout, Softwaredemonstration, Tafel |

Lernziele

Die Studierenden ...

- schätzen die Vorteile und Gefahren von Skriptsprachen in der Software-Entwicklung am Beispiel der Unix-Shells sh (dash) und bash ab.
- wenden reguläre Ausdrücke praktisch für die Verarbeitung von Texten und Auszeichnungssprachen an und kennen deren Mächtigkeit und Grenzen.
- arbeiten mit Filtern und Pipes und kombinieren diese zu einfachen Programmen beziehungsweise Skripten.
- haben ein Grundverständnis über interne Abläufe im Unix-Kern bei der Prozessverwaltung.
- haben ein Verständnis für die sequentielle Verarbeitung, wie Filterung und Auswertung, von großen Datenmengen in Textform.

Inhalt

- Unix
 - Systemstruktur
 - Shell Kommandos
 - Dateisystem und Rechteverwaltung
 - Filter und Pipelines
 - Skriptprogrammierung mit der Shell
 - POSIX-Konformität und nützliche Erweiterungen durch die bash
- Reguläre Ausdrücke
 - Aufbau und Zusammensetzung
 - Tools zur Verwendung von regulären Ausdrücken unter Unix (grep, sed)
 - Mächtigkeit und Grenzen von regulären Ausdrücken

- Suchen, Zerlegen und Editieren von Texten mit regulären Ausdrücken
- POSIX-konforme reguläre Ausdrücke und GNU-Erweiterungen
- Skriptsprachen
 - Einfache Shell-Programme (dash, bash)
 - Systematisches Kombinieren kleiner Programme
- Das make-System
- Prozessverwaltung

Literatur

- Karafiat, Helga: Unix & Shell-Programmierung Vorlesungsunterlagen im Web: <http://www.fh-wedel.de/~kar/unix-vorlesung/>
- Kofler, Michael: Linux: Das umfassende Handbuch, Rheinwerk Computing, 2017, ISBN: 978-38-36258-54-8
- Dietze, Martin: Praxiskurs Unix-Shell, O'Reilly Verlag GmbH & Co. KG; 2011, ISBN: 978-38-97215-65-8
- Robbins, Arnold; Beebe, Nelson H.F.: Klassische Shell-Programmierung, O'Reilly Verlag GmbH & Co. KG; 2006, ISBN: 978-38-97214-41-5
- Kernighan, Brian W.; Pike, Rob: UNIX-Werkzeugkasten: Programmieren mit UNIX, Hanser Fachbuch, 1986, ISBN-13: 978-34-46142-73-2
- Friedl, Jeffrey E. F.: Reguläre Ausdrücke, O'Reilly Verlag GmbH & Co. KG, 2007, ISBN: 978-38-97217-20-1
- Dougherty, Dale; Robbins, Arnold: sed & awk, O'Reilly and Associates, 1997, ISBN: 978-15-65922-25-9

I.1.24.2 Übg. UNIX & Shell-Programmierung

| | |
|---------------------------------|--|
| Lehrveranstaltung | Übg. UNIX & Shell-Programmierung |
| Dozent(en) | Malte Heins |
| Hörtermin | 4 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung mit integrierter Übung/Workshop/Assigm. |
| Semesterwochenstunden | 2 |
| ECTS | 3.0 |
| Prüfungsform | Abnahme |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, studentische Arbeit am Rechner, Tutorien |

Lernziele

Die Studierenden ...

- wenden praktisch die Inhalte aus der Vorlesung an.
- beherrschen den praktischen Umgang mit der Shell und den gängigsten UNIX Kommandos.
- erlangen die Fähigkeit zur Erstellung kleiner nichttrivialer Shell-Programme durch das Arbeiten mit Pipes und Filtern.

Inhalt

Bearbeitung von Übungsaufgaben parallel zum Stoff der Vorlesung in Zweiergruppen mit Abnahme und Diskussion der Lösungen.

Literatur

- Unterlagen zur Übung im Web
- siehe auch Vorlesung UNIX und Shell-Programmierung

I.1.25 Geschäftsprozesse mit ERP-Systemen

B253 Geschäftsprozesse mit ERP-Systemen

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B253 |
| Bezeichnung | Geschäftsprozesse mit ERP-Systemen |
| Lehrveranstaltung(en) | B253a Geschäftsprozesse mit ERP-Systemen B253b Übg. Geschäftsprozesse mit ERP-Systemen |
| Verantwortliche(r) | Dipl.-Inform. (FH) Birger Wolter |
| Zuordnung zum Curriculum | Betriebswirtschaftslehre (Bachelor) Data Science & Artificial Intelligence (Bachelor) E-Commerce (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) Wirtschaftsingenieurwesen (Bachelor) |
| Verwendbarkeit | Das Modul "Betriebswirtschaftliche Prozesse mit ERP-Systemen" baut auf erworbene Kenntnisse und Fähigkeiten aus grundlegenden betriebswirtschaftlichen Modulen wie "Einführung in die Betriebswirtschaft" und "Rechnungswesen 1 und 2" auf und vertieft diese im Hinblick auf eine umfassende Unterstützung von Geschäftsprozessen durch IT-Systeme. Das Modul lässt sich sinnvoll kombinieren mit den Modulen "Implementierung von Geschäftsprozessen in ERP-Systemen" und dem Modul "Systemmodellierung". |
| Semesterwochenstunden | 6 |
| ECTS | 5.0 |
| Voraussetzungen | Von den Studierenden wird ein grundlegendes Verständnis von betriebswirtschaftlichen Zusammenhängen in Unternehmen vorausgesetzt, wie es in den Vorsemestern vermittelt wird. |
| Dauer | 1 |
| Lernziele | Ohne den umfassenden Einsatz von ERP-Systemen wäre die effiziente Abwicklung von Geschäftsprozessen in heutigen Unternehmen nicht mehr vorstellbar. Nach Abschluss des Moduls verfügen die Studierenden daher über Wissen hinsichtlich der Realisierung der betriebswirtschaftlichen Grundfunktionalitäten innerhalb eines ERP-Systems. Insbesondere die integrierte |

Betrachtung von betriebswirtschaftlichen Vorgängen und Prozessen sowie deren informationstechnische Abbildung in ERP-Systemen steht dabei im Vordergrund. Durch die selbständige Bearbeitung von Fallstudien am ERP-System kann den Studenten die Bewältigung betriebswirtschaftlicher Vorgänge in einer praxisnahen Systemumgebung vermittelt werden und stellt somit eine gute Vorbereitung auf die berufliche Praxis dar.

I.1.25.1 Geschäftsprozesse mit ERP-Systemen

| | |
|---------------------------------|--|
| Lehrveranstaltung | Geschäftsprozesse mit ERP-Systemen |
| Dozent(en) | Fikret Koyuncu |
| Hörtermin | 4 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 2 |
| ECTS | 2.0 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, interaktive Entwicklung und Diskussion von Modellen, Softwaredemonstration |

Lernziele

Die Studierenden können ...

- wesentliche Begriffe im Umfeld von ERP-Systemen und Geschäftsprozessen definieren und diese in Beziehung zueinander setzen
- die wesentlichen Eigenschaften von ERP-Systemen und deren Architekturen herausstellen
- können die Vorgehensweise zur Abbildung betriebswirtschaftlicher Prozesse in ERP-Systemen erläutern.

Inhalt

Die Veranstaltung soll den Studierenden wichtige Grundlagen zum Verständnis und der Anwendung von ERP-Systemen in der Unternehmenspraxis vermitteln. Hierzu werden neben den begrifflichen Grundlagen und Inhalten zur Systemarchitektur von ERP-Systemen die wesentlichen Konzepte zur Abbildung von Geschäftsprozessen in ERP-Systemen vermittelt. Schwerpunkt bilden hierbei die Themen Daten- und Funktionsintegration aus einer prozessorientierten Perspektive.

Kurzgliederung:

- Grundlagen und Begriffe
- ERP-System-Architektur
- Integration von Geschäftsprozessen und ERP-Systemen
- Prozesse in der Produktion und deren Abbildung in ERP-Systemen
- Prozesse im Bereich Materialwirtschaft, Logistik und deren Abbildung in ERP-Systemen
- Prozesse im Bereich Controlling und deren Abbildung in ERP-Systemen
- Prozesse im Bereich Rechnungswesen und deren Abbildung in ERP-Systemen

Literatur

- Forsthuber, Heinz, Siebert, Jörg: Praxishandbuch SAP-Finanzwesen. Bonn; Boston, Mass : Galileo Press, 2013
- Frick, Detlev ; Gadatsch, A. ; Schäffer-Külz, U. G. (Hrsg.): Grundkurs SAP ERP: Geschäftsprozessorientierte Einführung mit durchgehendem Fallbeispiel. 2., Aufl. 2013. Aufl. Wiesbaden : Springer Fachmedien Wiesbaden GmbH, 2013
- Gadatsch, Andreas: Grundkurs Geschäftsprozess-Management: Methoden und Werkzeuge für die IT-Praxis: eine Einführung für Studenten und Praktiker. Wiesbaden : Vieweg + Teubner, 2012
- Mertens, Peter: Integrierte Informationsverarbeitung 1 Operative Systeme in der Industrie. Wiesbaden : Springer Fachmedien Wiesbaden: Imprint: Springer Gabler, 2013
- Weske, Mathias: Business process management concepts, languages, architectures. Berlin; New York : Springer, 2012

I.1.25.2 Übg. Geschäftsprozesse mit ERP-Systemen

| | |
|---------------------------------|---|
| Lehrveranstaltung | Übg. Geschäftsprozesse mit ERP-Systemen |
| Dozent(en) | Fikret Koyuncu |
| Hörtermin | 4 |
| Häufigkeit | jährlich |
| Lehrform | Übung/Praktikum/Planspiel |
| Semesterwochenstunden | 4 |
| ECTS | 3.0 |
| Prüfungsform | Abnahme |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Softwaredemonstration, studentische Arbeit am Rechner |

Lernziele

Die Studierenden können ...

- die erworbenen theoretischen Kenntnisse auf praktische Anwendungsfälle übertragen
- die grundlegende Funktionen von ERP-Systemen verwenden
- betriebswirtschaftliche Aufgabenstellungen und zusammenhängende Geschäftsvorfälle in verschiedenen Problemfeldern mit Hilfe von ERP-Systemen bearbeiten.

Inhalt

Die in der Vorlesung vermittelten theoretischen Grundlagen werden im Rahmen der Übung durch die Bearbeitung von Fallstudien durch eigene Anwendung gefestigt und erweitert. Die Bearbeitung der durchgängigen und ERP-modulübergreifenden Fallstudien erfolgt direkt am ERP-System am Beispiel des SAP ERP. Hiermit soll den Studierenden insbesondere eine integrierte Betrachtungsweise der betriebswirtschaftlichen Vorgänge und deren informationstechnische Abbildung in einem ERP-System anhand praktischer Übungen näher gebracht werden.

Kurzgliederung:

- Navigation im SAP ERP-System
- Fallstudie zu den Prozessen in der Produktion
- Fallstudie zu den Prozessen im Bereich Materialwirtschaft, Logistik
- Fallstudie zu den Prozessen im Bereich Controlling
- Zusammenhängendes Fallbeispiel zu den behandelten Themen

Literatur

KOYUNCU, Fikret: Übungsmaterialien zu den jeweiligen Fallstudien auf Moodle

I.1.26 Induktive Statistik

B041 Induktive Statistik

| | |
|---------------------------------|--|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B041 |
| Bezeichnung | Induktive Statistik |
| Lehrveranstaltung(en) | B041a Induktive Statistik |
| Verantwortliche(r) | Prof. Dr. Franziska Bönte |
| Zuordnung zum Curriculum | Betriebswirtschaftslehre (Bachelor) Computer Games Technology (Bachelor) Data Science & Artificial Intelligence (Bachelor) E-Commerce (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Smart Technology (Bachelor) Wirtschaftsinformatik (Bachelor) Wirtschaftsingenieurwesen (Bachelor) |
| Verwendbarkeit | Dieses Modul setzt Grundkenntnisse der Statistik, wie sie zum Beispiel in der Veranstaltung "Deskriptive Statistik" im Modul "Deskriptive Statistik & Grundlagen der Linearen Algebra" erworben werden, voraus. Die Kenntnisse aus dem Modul versetzen die Studierenden in die Lage quantitative Auswertung, wie sie zum Beispiel in empirischen Studien erforderlich sind, vorzunehmen. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Die Studierenden besitzen Grundkenntnisse der deskriptiven Statistik, wie sie in der Veranstaltung "Deskriptive Statistik" vermittelt werden. |
| Dauer | 1 |

Lernziele

Nach Abschluss des Moduls sind die Studierenden befähigt, weiterführende statistische Methoden zur Lösung komplexer Problemstellungen zu nutzen und die erzielten Ergebnisse korrekt interpretieren zu können.

Die Studierenden erwerben fundierte Kenntnisse in den Methoden der schließenden Statistik. Sie sind in der Lage komplexe, statistische Untersuchungen - von der Datenerhebung bis

zur Auswertung und Interpretation - auch unter Zuhilfenahme geeigneter Computerprogramme eigenständig vorzunehmen und zu bewerten.

Sie kennen die grundlegenden Methoden der statistischen Qualitätskontrolle und wissen, wie Testverfahren im Rahmen der statistischen Qualitätskontrolle genutzt werden können.

Die Studenten sind in der Lage, Zeitreihen zu analysieren und unter Anwendung statistischer Verfahren Prognosen zu erstellen.

I.1.26.1 Induktive Statistik

| | |
|---------------------------------|---|
| Lehrveranstaltung | Induktive Statistik |
| Dozent(en) | Franziska Bönte |
| Hörtermin | 5 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, E-Learning, Handout, Online-Aufbereitung, studentische Arbeit am Rechner, Tafel, Tutorien |

Lernziele

Die Studierenden sind befähigt, weiterführende statistische Methoden zur Lösung komplexer Problemstellungen nutzen und die erzielten Ergebnisse korrekt zu interpretieren.

Die Studierenden erlangen ...

- Verteilungsannahmen für unterschiedliche ökonomische und naturwissenschaftliche Fragestellungen sinnvoll zu tätigen
- die Fähigkeit, Schätzwerte für die Parameter einer Grundgesamtheit zu bestimmen.
- die Fähigkeit, selbständig statistische Tests im Rahmen betrieblicher Aufgabenstellungen zu planen und durchzuführen und die Ergebnisse korrekt anzugeben.
- die Fähigkeit, Intervallwahrscheinlichkeiten unter Verwendung der wichtigsten diskreten und stetigen Dichte- und Verteilungsfunktionen zu berechnen.
- die Fähigkeit, Werte einer Grundgesamtheit zu schätzen und Hypothesen über die Werte einer Grundgesamtheit zu testen.
- die Fähigkeit, mittels geeigneter Computerprogramme statistische Untersuchungen großer Datenmengen vorzunehmen.
- Kenntnisse hinsichtlich des Einsatzes von Testverfahren im Rahmen der statistischen Qualitätskontrolle anhand von Problemstellungen aus der Wirtschaft.
- die Fähigkeit, sowohl eine Zeitreihe zu analysieren und die Komponenten einer Zeitreihe zu berechnen als auch kurz- und langfristige Prognosen durchzuführen.
- die Fähigkeit, die Genauigkeit von Prognosen kritisch zu bewerten.

Inhalt

- Wahrscheinlichkeitsrechnung
 - Grundlagen
 - Zufallsvariable und Wahrscheinlichkeitsverteilungen

- Kombinatorik
- Ausgewählte diskrete Verteilungen
- Ausgewählte stetige Verteilungen
- Hauptsätze der Wahrscheinlichkeitstheorie und Statistik
- Approximationsregeln
- Stichproben
- Schätzverfahren
 - Punktschätzung
 - Intervallschätzung
- Testverfahren
 - Parametrische Testverfahren
 - Verteilungstests

Literatur

- Bley Müller, Josef: Statistik für Wirtschaftswissenschaftler; 16. Aufl.; s.l.; Verlag Franz Vahlen; 2012
- Griffiths, William E.; Hill, R. Carter; Judge, George G.: Learning and practicing econometrics; ; New York; John Wiley; 1993
- Hansen, Gerd: Methodenlehre der Statistik; ; München; Vahlen; 1974
- Hansmann, Karl-Werner: Kurzlehrbuch Prognoseverfahren; ; Wiesbaden; s.l.; Gabler Verlag; 1983
- Lippe, Peter Michael von der: Wirtschaftsstatistik; 3., neubearb. u. erw. Aufl.; Stuttgart; Fischer; 1985
- Mood, Alexander MacFarlane; Boes, Duane C.; Graybill, Franklin A.: Introduction to the theory of statistics; 3. ed., international ed., [reprint.]; Auckland; McGraw-Hill; 2009
- Rüger, Bernhard: Induktive Statistik; 2., überarb. Aufl., 2. Nachdr; München; Oldenbourg; 1995
- Schlittgen, Rainer; Streitberg, Bernd H. J.: Zeitreihenanalyse; 3. Aufl., durchges. u. verb; München; R. Oldenbourg; 1989
- Zuckarelli, Joachim: Statistik mit R; ; Heidelberg; O'Reilly; 2017
- Bourier, Günther: Beschreibende Statistik. 11. Auflage. Wiesbaden: Springer Gabler, 2013.
- Kobelt, Helmut; Steinhausen, Detlef: Wirtschaftsstatistik für Studium und Praxis. 7. Auflage. Stuttgart: Schäfer-Poeschel Verlag, 2006.
- Schwarze, Jochen: Grundlagen der Statistik Band 2 : Wahrscheinlichkeitsrechnung und induktive Statistik. 10. Auflage. Berlin: nwb Studium 2013.

- Toutenburg, Helge u., a.: Induktive Statistik : Eine Einführung mit R und SPSS. 4. Auflage. Berlin: Springer-Verlag 2008.

I.1.27 Anwendungen der Künstlichen Intelligenz

B095 Anwendungen der Künstlichen Intelligenz

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B095 |
| Bezeichnung | Anwendungen der Künstlichen Intelligenz |
| Lehrveranstaltung(en) | B095a Anwendungen der Künstlichen Intelligenz |
| Verantwortliche(r) | Prof. Dr. Sebastian Iwanowski |
| Zuordnung zum Curriculum | Computer Games Technology (Bachelor) Data Science & Artificial Intelligence (Bachelor) E-Commerce (Bachelor) IT-Ingenieurwesen (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Smart Technology (Bachelor) Technische Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) |
| Verwendbarkeit | Das Modul liefert praktische Anwendungen aus verschiedenen Bereichen der teilnehmenden Studiengänge. Es gibt Ideen für das Praktikum und die anschließende Bachelor-Thesis. Es liefert Grundlagen, die zur Aufnahme eines Masterstudiums motivieren. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Erwartet werden Kenntnisse der Diskreten Mathematik sowie gute Programmierkenntnisse. Die Teilnehmer sollten bereits größere Programme geschrieben haben und Problemstellungen aus der Praxis kennengelernt haben (mindestens im Rahmen angewandter Vorlesungen). Vertrautheit mit objektorientierter Programmierung ist von Vorteil. |
| Dauer | 1 |

Lernziele

Nach Abschluss des Moduls kennen die Studierenden die grundsätzlichen Ziele der Künstlichen Intelligenz und deren Anwendungsbezug. Die Studierenden kennen komplexe Anwendungsbeispiele und sind in der Lage, elementare Techniken der Künstlichen Intelligenz in Implementierungen anzuwenden. Hierfür verfügen sie über eine grundlegende Kenntnis wichtiger Basistechnologien der Künstlichen Intelligenz.

I.1.27.1 Anwendungen der Künstlichen Intelligenz

| | |
|---------------------------------|--|
| Lehrveranstaltung | Anwendungen der Künstlichen Intelligenz |
| Dozent(en) | Sebastian Iwanowski |
| Hörtermin | 5 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung mit integrierter Übung/Workshop/Assig. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Prüfungsform | Klausur + ggf. Bonus |
| Sprache | deutsch/englisch |
| Lehr- und Medienform(en) | |

Lernziele

Nach Abschluss der Veranstaltung besitzen die Studierenden folgende Kompetenzen:

- Kenntnis und Interesse für die grundsätzlichen Ziele der Künstlichen Intelligenz.
- Kenntnis der Basistechnologien der Künstlichen Intelligenz.
- Fähigkeit, elementare Techniken der Künstlichen Intelligenz in Implementierungen anzuwenden.
- Kenntnis verschiedener komplexer Anwendungsbeispiele.

Inhalt

- Einführung
 - Definition und Ziele der KI
 - Überblick über die Basistechnologien der KI
 - Auswahl von Anwendungsbeispielen
- Basistechnologien
 - Wissensbasierte Systeme mit den Ausprägungen Regelbasierte Systeme, Modellbasierte Systeme und Fallbasierte Systeme
 - Machine Learning
 - Suchstrategien
 - Schwarmintelligenz
 - Grundlagen von semantischen Netzwerken
- Anwendungen
 - Verkehrsinformation und -navigation
 - Logistische Fragestellungen
 - Technische Diagnose

- Bilderkennung
- Spiele

Literatur

- Marco Dorigo / Thomas Stützle:
Ant Colony Optimization,
MIT Press 2004, ISBN 0-262-04219-3
- Goodfellow, Ian, Yoshua Bengio und Aaron Courville: Deep Learning. <http://www.deeplearningbook.org>. MIT Press, 2016. ISBN: 978-0-262-03561-3
- Ute Schmid / Günter Görz / Josef Schneeberger:
Handbuch der Künstlichen Intelligenz,
Oldenbourg 2013 (5. Auflage), ISBN 978-3-486-71307-7
- Stuart Russell / Peter Norvig:
Künstliche Intelligenz: Ein moderner Ansatz,
Pearson Studium 2012 (3. Auflage), ISBN 978-3-86894-098-5
- Liyang Yu : A Developer's Guide to the Semantic Web , Springer 2011, ISBN 978-3-642-15969-5

I.1.28 Softwarequalität

B093 Softwarequalität

| | |
|---------------------------------|--|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B093 |
| Bezeichnung | Softwarequalität |
| Lehrveranstaltung(en) | B093a Softwarequalität |
| Verantwortliche(r) | Prof. Dr. Gerd Beuster |
| Zuordnung zum Curriculum | E-Commerce (Bachelor) IT-Ingenieurwesen (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Wirtschaftsinformatik (Bachelor) |
| Verwendbarkeit | Das Modul setzt grundlegende Kenntnisse der Softwareerstellung, insbesondere der Programmierung in einer höheren Programmiersprache, voraus. Die Studierenden sind mit den Kenntnissen des Moduls in der Lage, allgemeine Methoden der Qualitätssicherung auf Software anzuwenden. Sie sind auch mit den speziellen Methoden der Qualitätssicherung von Software, insbesondere Methoden zur systematischen Erstellung von Softwaretests, vertraut. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Kenntnis grundlegender Eigenschaften von Software-Systemen und ihrer Realisierung, Fähigkeit zur Analyse von Software, um ein Verständnis der darin enthaltenen Zusammenhänge zu erlangen, Kenntnisse hinsichtlich der Vernetzung von Rechnern und der Software-Konzepte zur Nutzung der Vernetzung. |
| Dauer | 1 |

Lernziele

Nach Abschluss des Moduls verfügen die Studierenden über Kenntnisse hinsichtlich verschiedener Qualitätsaspekte von Software-Systemen. Die Studierenden kennen Qualitätsmerkmale und -kriterien und die Methoden zur Erreichung entsprechender Qualitätsziele.

Die Studierenden wissen, welche Bedeutung weitere Qualitätsmerkmale von Software haben.

Darüber hinaus kennen sie die Gründe für das Zustandekommen von Qualitätsdefizite und die Maßnahmen zur Gewährleistung eines geforderten Qualitätsniveaus.

Die Studierenden haben systematisches Testen als Mittel zur Qualitätssicherung und -kontrolle kennengelernt. Sie können die gängigen Methoden und Verfahren zum White-Box-Testing (Testen unter Kenntnis der Spezifikation und/oder Implementierung) und Black-Box-Testing (Testen ohne Kenntnis der internen Funktionsweise des IT-Systems) theoretisch begründen und praktisch umsetzen.

I.1.28.1 Softwarequalität

| | |
|---------------------------------|---|
| Lehrveranstaltung | Softwarequalität |
| Dozent(en) | Jochen Brunnstein |
| Hörtermin | 5 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung mit integrierter Übung/Workshop/Assigm. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Prüfungsform | Klausur + ggf. Bonus |
| Sprache | deutsch/englisch |
| Lehr- und Medienform(en) | |

Lernziele

Die Studierenden kennen Qualitätsmerkmale und -kriterien und die Methoden zur Erreichung entsprechender Qualitätsziele. Sie können die gängigen Methoden und Verfahren zur Sicherstellung von Softwarequalität umsetzen.

- Kenntnis der wesentlichen Qualitätsmerkmale von Software und ihrer wechselseitigen Abhängigkeiten.
- Kenntnis der typischen Defizite der Software-Qualität und ihrer Gründe.
- Kenntnis der Aufgabenbereiche des Software-Qualitätsmanagement und Überblick über die wesentlichen Managementkonzepte.
- Kenntnis des Konzepts der Qualitätsmodelle und der relevanten Qualitätsmerkmale und -metriken.
- Überblick über mögliche Maßnahmen der Software-Qualitätssicherung, Kenntnis der wesentlichen konstruktiven und analytischen Maßnahmen der Software-Qualitätssicherung.
- Fähigkeit, ausgewählte Maßnahmen der Qualitätssicherung umzusetzen.
- Erkenntnis der besonderen Bedeutung der Usability als benutzerzentriertes Qualitätsmerkmal.
- Kenntnis der wesentlichen Ansätze, die Usability einer Software zu bewerten und zu gestalten.
- Kenntnis der methodischen Ansätze zur angemessenen Einbeziehung ergonomischer Aspekte in Software-Entwicklungsprozesse, insbesondere Potentiale und Probleme partizipativer Software-Entwicklung.

Inhalt

- Einführung und Motivation
 - Definition des Begriffs "Software-Qualität"
 - Bedeutung der Software-Qualität
- Merkmale der Software-Qualität

- Software-Maße und -Metriken
- Modelle der Software-Qualität
- Einschränkungen der Software-Qualität und ihre Gründe
- Software-Qualitätsmanagement
 - Aufgabenbereiche
 - Grundlegende Prinzipien
- Maßnahmen der Software-Qualitätssicherung
 - Konstruktive Maßnahmen
 - Prozessbezogene Maßnahmen
 - Produktbezogene Maßnahmen
- Analytische Maßnahmen
 - Statische Prüftechniken
 - Dynamische Prüftechniken
- Testen als Maßnahme der Qualitätssicherung
- Black-Box- und White-Box-Testing
- Verfahren des Black-Box-Testing
- Verfahren des White-Box-Testing
 - Graphenbasierte Testfallgenerierung
 - Schnittstellensignaturbasierte Testfallgenerierung
 - Testfallgenerierung nach logischen Kriterien
 - Syntaxbasierte Testfallgenerierung
- Testen eingebetteter Systeme

Literatur

- Ammann, Paul; Offutt, Jeff: Introduction to Software Testing. 1. Auflage. Cambridge, UK: Cambridge University Press, 2008.
- Balzert, Helmut: Lehrbuch der Softwaretechnik : Basiskonzepte und Requirements Engineering. 3. Auflage. Heidelberg: Spektrum Akademischer Verlag, 2009.
- Balzert, Helmut: Lehrbuch der Softwaretechnik : Softwaremanagement. 2. Auflage. Heidelberg: Spektrum Akademischer Verlag, 2008.
- Hoffmann, Dirk W.: Software-Qualität. Berlin: Springer-Verlag, 2008.
- Kneuper, Ralf: CMMI : Verbesserung von Software- und Systementwicklungsprozessen mit Capability Maturity Model Integration. Heidelberg: Dpunkt Verlag, 2007
- Kahn, Stephen H.: Metrics and Models in Software Quality Engineering. 2. Auflage. Boston (MA), USA: Addison-Wesley, 2002.

- Liggesmeyer, Peter: Software-Qualität : Testen, Analysieren und Verifizieren von Software. 2. Auflag. Heidelberg: Spektrum Akademischer Verlag, 2009.
- Schneider, Kurt: Abenteuer Softwarequalität : Grundlagen und Verfahren für Qualitätssicherung und Qualitätsmanagement. 2. Auflage. Heidelberg: Dpunkt Verlag, 2012
- Tian, Jeff : Software Quality Engineering. 1. Auflage. Hoboken (NJ), USA: John Wiley & Sons, 2005.
- Wallmüller, Ernest: Software Quality Engineering : Ein Leitfaden für bessere Software-Qualität. 3. Auflage. München: Carl Hanser Verlag, 2011.

I.1.29 Systemmodellierung

B087 Systemmodellierung

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B087 |
| Bezeichnung | Systemmodellierung |
| Lehrveranstaltung(en) | B087a Systemanalyse B087b Prozessmodellierung |
| Verantwortliche(r) | Prof. Dr. Andreas Häuslein |
| Zuordnung zum Curriculum | Data Science & Artificial Intelligence (Bachelor) E-Commerce (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) Wirtschaftsingenieurwesen (Bachelor) |
| Verwendbarkeit | Das Modul baut auf grundlegenden Kenntnissen der Programmierung auf (zum Beispiel "Programmstrukturen 1" oder "Einführung in die Programmierung"). Da es die Sichtweise auf die Programmierung auf die vorgelagerten Phasen der Programmierung im engeren Sinne ausweitet, bildet es Kompetenzen aus, die in allen Modulen verwendbar sind, in denen die Ermittlung von fachlichen Anforderungen als Grundlage einer anschließenden Systementwicklung erforderlich ist. Beispielsweise sind dies die Module "Prozessmodellimplementation", "Geschäftsprozesse mit ERP-Systemen", "Software-Projekt" und "Bachelor-Thesis". Das Modul ist sowohl in Informatik- als auch in Wirtschaftsstudiengängen verwendbar. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Kenntnisse der grundlegenden Vorgehensweise bei der Software-Entwicklung, Wissen hinsichtlich der Grundkonzepte von Programmiersprachen, Kenntnisse bezogen auf die Grundfunktionen eines Unternehmens und seinen Aufbau, Fähigkeit zur Abstraktion |
| Dauer | 1 |

Lernziele

Das Modul dient der Vermittlung einer gleichermaßen fach- wie systembezogenen Sicht auf Anwendungs- und entsprechende Software-Systeme. Ein Schwerpunkt ist der Erwerb von Kenntnis der wesentlichen Entwicklungstätigkeiten und ihrer methodischen Grundlagen, die der Implementierung von Software vorgelagert sind.

Die Studierenden erlangen ein Einschätzungsvermögen hinsichtlich der Notwendigkeit und Grenzen von Systemanalysen, insbesondere in Bezug auf Analyse von Geschäftsprozessen und die entsprechende Gestaltung betrieblicher Informationssysteme. Sie erwerben Kenntnisse der wesentlichen Techniken zur Informationsgewinnung in Unternehmen einschließlich ihrer Vor- und Nachteile. Sie kennen die im Unternehmensumfeld praktisch relevanten methodischen Ansätze zur System- und Prozessmodellierung und die damit verbundenen Modellnotationen.

Sie besitzen die Fähigkeit zur Nutzung der Modellierungsmittel zum Aufbau von Analysemodellen für wirtschaftliche Problemstellungen mittlerer Komplexität. Sie können somit an der Ermittlung von fachlichen Anforderungen für eine Systementwicklung mitwirken und sind in der Lage, eine Systemspezifikation als Ausgangspunkt einer solchen Entwicklung zu erstellen. Sie besitzen die Fähigkeit, eine prozessorientierte Sichtweise auf die Abläufe in einem Unternehmen einzunehmen und die Geschäftsprozesse in Form von Modellen abzubilden, auch unter Einsatz geeigneter Modellierungssoftware.

I.1.29.1 Systemanalyse

| | |
|---------------------------------|--|
| Lehrveranstaltung | Systemanalyse |
| Dozent(en) | Andreas Häuslein |
| Hörtermin | 5 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 2 |
| ECTS | 3.0 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, E-Learning, Gastreferenten, Handout, interaktive Entwicklung und Diskussion von Modellen, Online-Aufbereitung, Overheadfolien, Tafel |

Lernziele

Die Studierenden ...

- beurteilen die generellen Möglichkeiten und Grenzen von Systemanalysen, insbesondere in Bezug auf die Gestaltung betrieblicher Informationssysteme.
- unterscheiden die wesentlichen Techniken zur Informationsgewinnung in Unternehmen einschließlich ihrer Vor- und Nachteile, bewerten Techniken im Kontext einer konkreten Informationsgewinnung.
- führen eine methodisch fundierte Informationsgewinnung in einem überschaubaren Problemkontext durch.
- erklären wichtige Bestandteile und Schritte der Systemaufnahme als Vorphase zur Systemmodellierung, sie setzen ausgewählte Formalismen zur Dokumentation der Aufnahmeergebnisse ein.
- stellen die im Unternehmensumfeld praktisch relevanten methodischen Ansätze zur Systemmodellierung dar und beurteilen diese hinsichtlich ihrer Eignung für bestimmte Erkenntnisziele im Rahmen einer Systemanalyse.
- beschreiben die zu den methodischen Ansätzen gehörenden Modellnotationen und setzen diese angemessen zur Modellierung ein.
- nutzen die Modellierungsmittel zum Aufbau von Analysemodellen begrenzter Komplexität für betriebswirtschaftlich ausgerichtete Informationssysteme und diesbezügliche Problemstellungen.

Inhalt

- Grundbegriffe der Systemanalyse
 - Gegenstand und Zielsetzung im Unternehmensumfeld
 - Methodische Grundlagen
- Systemaufnahme

- Rahmenbedingungen und Techniken der Informationsgewinnung
- Thematische Untersuchungsbereiche
- Systemmodellierung
 - Ereignisgesteuerte Prozessketten zur Modellierung von Geschäftsprozessen
 - * Modellelemente schlanker EPK-Modelle und ihre Nutzung
 - * Modellelemente erweiterter EPK-Modelle und ihre Nutzung
 - Business Process Model and Notation BPMN
 - * Ausgewählte Modellelemente
 - * Beispielmodelle
 - Objektorientierte Analyse
 - * Statische Modelle
 - * Dynamische Modelle
 - Strukturierte Analyse und Essenzielle Modellierung
 - * Darstellungs- und Modellierungsmittel
 - * Ausgewählte Schritte des Vorgehensmodells

Literatur

- KRALLMANN, H.; BOBRIK, A.; LEVINA, O.: Systemanalyse im Unternehmen - Prozessorientierte Methoden der Wirtschaftsinformatik, Oldenbourg, 2013
- ALLWEYER, Thomas: BPMN 2.0 - Business Process Model and Notation: Einführung in den Standard für die Geschäftsprozessmodellierung. 2. Auflage, Books on Demand, 2020
- GADATSCH, Andreas: Grundkurs Geschäftsprozess-Management: Analyse, Modellierung, Optimierung und Controlling von Prozessen. 9. Aufl., Springer Vieweg, 2020
- RUPP, Chr.: Systemanalyse kompakt, Springer Verlag, 2013
- HÄUSLEIN, A.: Systemanalyse. vde-Verlag, 2004
- KRÜGER, J.; UHLIG, Ch.: Praxis der Geschäftsprozessmodellierung. VDE Verlag, 2009
- SCHEER, A.-W.: Architektur integrierter Informationssysteme. Springer-Verlag, Berlin, 1991
- OBJECT MANAGEMENT GROUP OMG: Business Process Model and Notation (BPMN), Version 2.0, URL: <http://www.omg.org/spec/BPMN/2.0>, 2011
- FREUND, Jakob ; RÜCKER, Bernd: Praxishandbuch BPMN 2.0. 4. aktualisierte Auflage, Carl Hanser Verlag, 2014
- OESTERREICH, B.: Analyse und Design mit UML 2.3: Objektorientierte Softwareentwicklung. Oldenbourg, 2009
- KECHER, Chr.: UML 2: Das umfassende Handbuch. Galileo Computing, 2011

I.1.29.2 Prozessmodellierung

| | |
|---------------------------------|---|
| Lehrveranstaltung | Prozessmodellierung |
| Dozent(en) | Christian Uhlig |
| Hörtermin | 5 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung mit integrierter Übung/Workshop/Assigm. |
| Semesterwochenstunden | 2 |
| ECTS | 2.0 |
| Prüfungsform | Abnahme |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, Handout, interaktive Entwicklung und Diskussion von Modellen, Softwaredemonstration, studentische Arbeit am Rechner |

Lernziele

Die Studierenden ...

- nennen und erläutern die theoretischen Grundlagen des Geschäftsprozessmanagement und seine Begriffswelt.
- nennen und erläutern die Grundlagen der Geschäftsprozessmodellierung sowie ihre Einordnung in das Geschäftsprozessmanagement und andere Themenbereiche wie Softwareengineering, Datenbanken und Systemanalyse.
- stellen die Grundlagen der ARIS-Methode dar.
- wenden wesentliche Modellierungskonzepte des ARIS-Softwaresystems (insbesondere Objekte und Kanten, Definitions- und Ausprägungsebenen sowie Hinterlegungen) an.
- nutzen das ARIS-Softwaresystem in seinen wesentlichen Bedienkonzepten und -elementen zur Erstellung von miteinander vernetzten Modellen.
- analysieren komplexe textuelle Fachkonzeptbeschreibungen und unterscheiden dabei Inhalte der verschiedenen ARIS-Modellierungssichten (Organisation, Daten, Leistungen, Funktionen, Steuerung).
- entwerfen und gestalten in ästhetisch ansprechender Weise Modelle zentraler Modelltypen (ER-Modell, EPK, WSK, BPMN Process und Collaboration, Organigramm) zu komplexen Fachkonzeptbeschreibungen.

Inhalt

- Grundlagen des Geschäftsprozessmanagement
 - Motivation
 - Begriffe
 - Einordnung der Geschäftsprozessmodellierung
 - Bezüge zur Systemanalyse und zum Software-Engineering
- ARIS-Methode

- Sichtenkonzept
- Schichtenkonzept
- Überblick über Modelltypen und ihre Vernetzung
- ARIS-Softwaresystem
- Modellierung der Aufbauorganisation
- ER-Datenmodellierung
- Funktionsmodellierung
- Prozessmodellierung
 - Wertschöpfungsketten (WSK) und Prozesslandkarten
 - EPK
 - Vernetzung mit anderen ARIS-Sichten (Daten, Aufbauorganisation)
 - BPMN (Process und Collaboration Diagrams)
- Praktische Aufgabenstellungen
 - Ausschnittsweise und formfreie Modellierung von Prozessen aus einem beispielhaften Fachkonzept
 - Modellierung des Datenmodells zu einem beispielhaften Fachkonzept (ERD)
 - Ausschnittsweise Modellierung von Prozessen zu einem beispielhaften Fachkonzept (WSK / EPK und BPMN)
 - Ganzheitliche Modellierung von Aufbauorganisation, Daten und Prozessen zu einer Fallstudie (Organigramm, ERD, WSK, BPMN)

Literatur

- Krüger, Jörg; Uhlig, Christian: Praxis der Geschäftsprozessmodellierung - ARIS erfolgreich anwenden, VDE Verlag, 2009
- Lehmann, Frank: Integrierte Prozessmodellierung mit ARIS, dpunkt.verlag, 2007
- Scheer, August-Wilhelm: ARIS Vom Geschäftsprozess zum Anwendungssystem, 4. Auflage, Springer, 2002
- Scheer, August-Wilhelm: ARIS Modellierungsmethoden, Metamodelle, Anwendungen, 4. Auflage, Springer, 2001
- OMG: Business Process Model and Notation (BPMN) Version 2.0, 2011
- Software AG: ARIS-Dokumentation (Methodenhandbuch, Bedienhandbücher), jeweils aktuellste Fassung

I.1.30 Seminar Informatik

B255 Seminar Informatik

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B255 |
| Bezeichnung | Seminar Informatik |
| Lehrveranstaltung(en) | B255a Seminar Informatik |
| Verantwortliche(r) | jeweiliger Dozent |
| Zuordnung zum Curriculum | Informatik (Bachelor) |
| Verwendbarkeit | Das Modul bildet eine sinnvolle Vorbereitung auf die Bachelor-Thesis. |
| Semesterwochenstunden | 2 |
| ECTS | 5.0 |
| Voraussetzungen | Veranstaltungsinhalte der ersten vier Studiensemester |
| Dauer | 1 |

Lernziele

Ziel des Moduls Seminar ist, die Fähigkeit zu erwerben, sich eigenständig in ein anspruchsvolles Informatik-Thema einzuarbeiten zu können und dieses geeignet, sowohl im Rahmen eines Vortrags als auch in Form einer schriftlichen Ausarbeitung darstellen zu können. Als Themen werden dabei aktuelle Entwicklungen in der angewandten Informatik aber auch grundlegende Themen der theoretischen Informatik gewählt. Studierende erlernen, gezielte Literaturrecherchen, insbesondere unter Berücksichtigung der Quellen des Internets, durchzuführen. Sie entwickeln und optimieren ihre Fähigkeiten zur Präsentation des Themas in freien Vorträgen, beim Umgang mit Präsentationsmedien und durch die offene Diskussion wissenschaftlicher Themen in der Gruppe. Die Erstellung einer stilistisch und fachlich ansprechenden Ausarbeitung dient der Vorbereitung auf die Bachelor-Arbeit.

I.1.30.1 Seminar Informatik

| | |
|---------------------------------|--|
| Lehrveranstaltung | Seminar Informatik |
| Dozent(en) | jeweiliger Dozent |
| Hörtermin | 5 |
| Häufigkeit | jedes Semester |
| Lehrform | Seminar |
| Semesterwochenstunden | 2 |
| ECTS | 5.0 |
| Prüfungsform | Schriftl. Ausarbeitung (ggf. mit Präsentation) |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, Handout, interaktive Entwicklung und Diskussion von Modellen, Online-Aufbereitung, Software-demonstration, studentische Arbeit am Rechner, Tafel |

Lernziele

Seminararbeiten dienen insbesondere dem Erlernen von Fertigkeiten zum Erstellen der Bachelor-Thesis. Wesentlich ist die eigenständige Erarbeitung und Darlegung der Inhalte zu einem vorgegebenen Thema unter Einhaltung der Formalia. Die Ausarbeitung soll das Interesse an einer eigenständigen Befassung mit Inhalten aus dem Themengebiet und den Einstieg in die zugehörige wissenschaftliche Fachliteratur und Methodik fördern und anregen. Schließlich ist die obligatorische Präsentation der Ergebnisse ebenfalls Aufgabe innerhalb des Seminars.

Nach erfolgreicher Teilnahme können sie ...

- wissenschaftliche Themen angemessen strukturieren.
- eine eigenständige Zielsetzung erarbeiten und umsetzen.
- Inhalte recherchieren und übersichtlich aufbereiten.
- formale Kriterien sicher beachten und anwenden.
- eine schriftliche Ausarbeitung größeren Umfangs erstellen.
- kontroverse Lehrmeinungen und aktuelle Trends zu einem Thema herausarbeiten.
- ihre Ergebnisse in angemessener Form vortragen und mit den Seminarteilnehmern diskutieren.

Inhalt

Fachvorträge mit anschließender Gruppendiskussion.

Literatur

Recherche nach aufgabenbezogener Literatur, teilweise aufgabenspezifische Vorgabe einzelner Literaturquellen.

Empfehlungen zur Einführung in das wissenschaftliche Arbeiten

- Axel Bänsch, Dorothea Alewell, Wissenschaftliches Arbeiten, 11. Aufl., München [u.a.]: Oldenbourg 2013.
- Werner Heister, Dagmar Weßler-Poßberg, Studieren mit Erfolg: Wissenschaftliches Arbeiten für Wirtschaftswissenschaftler, 2., überarbeitete Auflage, Stuttgart: Schäffer-Poeschel 2011.
- Jens Hiller, Arbeitstechniken und wissenschaftliches Arbeiten, Herne: Kiehl 2017.
- Walter Krämer, Wie schreibe ich eine Seminar- oder Examensarbeit. 3., überarbeitete und aktualisierte Aufl., Frankfurt: Campus 2009.
- Lydia Prexl, Mit digitalen Quellen arbeiten. Richtig zitieren aus Datenbanken, E-Books, YouTube & Co., 2., aktualisierte und erweiterte Aufl., Paderborn: Ferdinand Schöningh (UTB) 2016.
- Manuel René Theisen, Wissenschaftliches Arbeiten: Technik - Methodik Form, 15. Aufl., München: Vahlen 2011.

I.1.31 Anwendungsentwicklung in ERP-Systemen

B098 Anwendungsentwicklung in ERP-Systemen

| | |
|---------------------------------|--|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B098 |
| Bezeichnung | Anwendungsentwicklung in ERP-Systemen |
| Lehrveranstaltung(en) | B098b Übg. Anwendungsentwicklung in ERP-Systemen B098a Anwendungsentwicklung in ERP-Systemen |
| Verantwortliche(r) | Dipl.-Inform. (FH) Birger Wolter |
| Zuordnung zum Curriculum | IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) |
| Verwendbarkeit | Das Modul "Anwendungsentwicklung in ERP-Systemen" baut auf die unter anderem in den Modulen "Programmstrukturen 1" und "Einführung in Datenbanken" erworbenen Kenntnisse und Fähigkeiten auf und ergänzt die betriebswirtschaftliche Sichtweise des Moduls "Geschäftsprozesse mit ERP-Systemen" um die programmiertechnische Sichtweise. |
| Semesterwochenstunden | 6 |
| ECTS | 5.0 |
| Voraussetzungen | Programmstrukturen und Datenbankgrundlagen. |
| Dauer | 1 |

Lernziele

Nach Abschluss des Moduls verfügen die Studierenden über grundlegende und weiterführende Kenntnisse von Anwendungssystemen zur Planung und Steuerung betrieblicher Ressourcen (ERP-Systeme) - insbesondere der Programmierung und dem Customizing. Praktisch beherrschen die Studierenden das ERP-System der SAP AG bzw. die Programmiersprache ABAP sowie Webtechnologien wie Web Dynpro und SAPUI5.

I.1.31.1 Übg. Anwendungsentwicklung in ERP-Systemen

| | |
|---------------------------------|---|
| Lehrveranstaltung | Übg. Anwendungsentwicklung in ERP-Systemen |
| Dozent(en) | Birger Wolter |
| Hörtermin | 5 |
| Häufigkeit | jährlich |
| Lehrform | Übung/Praktikum/Planspiel |
| Semesterwochenstunden | 4 |
| ECTS | 3.0 |
| Prüfungsform | Abnahme |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Softwaredemonstration, studentische Arbeit am Rechner |

Lernziele

Die Studierenden können - durch Programmierübungen mit den Werkzeugen der Entwicklungsumgebung - die praktische Umsetzung der theoretisch erworbenen Kenntnisse.

Inhalt

- Datenbankzugriffe
- Listenverarbeitung (Reports)
- Gestaltung von Bildelementen (Dynpros)
- Web Dynpro und SAPUI5

Literatur

- WOLTER, Birger: Übungsmaterialien auf dem Handout-Server. <http://www.fh-wedel.de/mitarbeiter/wol/veranstaltungen/erp-systeme/> (kontinuierliche Aktualisierung)

I.1.31.2 Anwendungsentwicklung in ERP-Systemen

| | |
|---------------------------------|---|
| Lehrveranstaltung | Anwendungsentwicklung in ERP-Systemen |
| Dozent(en) | Birger Wolter |
| Hörtermin | 5 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung mit integrierter Übung/Workshop/Assigm. |
| Semesterwochenstunden | 2 |
| ECTS | 2.0 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Softwaredemonstration |

Lernziele

Die Studierenden ...

- erwerben systematische Kenntnisse der ABAP-Programmiersprache, als bedeutender Träger der betriebswirtschaftlichen Anwendungslogik in SAP-Systemen.
- erlangen die Fähigkeit, SAP-Systeme zu verstehen und einfache Erweiterungen für spezielle Anwendungsaufgaben vorzunehmen.

Inhalt

- Einführung und erste Schritte
- Werkzeuge der Entwicklungsumgebung
- Grundlegende Konzepte
- Datenbankzugriffe
- Listenverarbeitung (Reports)
- Gestaltung von Bildelementen (Dynpros)
- Besonderheiten von Unternehmenssoftware
- Objektorientierte Programmierung mit ABAP Objects
- Webtechnologien: Web Dynpro und SAPUI5

Literatur

- KELLER, Horst; KRÜGER, Sascha: ABAP Objects. ABAP-Programmierung mit SAP NetWeaver. Rheinwerk Verlag vorm. Galileo Press, 3. Auflage 2006
- KELLER, Horst; THÜMMEL, Wolf Hagen: ABAP - Das umfassende Handbuch. Rheinwerk Verlag vorm. Galileo Press, 1. Auflage 2014
- WOLTER, Birger: Veranstaltungsmaterialien auf dem Handout-Server. <http://www.fh-wedel.de/mitarbeiter/wol/veranstaltungen/erp-systeme/> (kontinuierliche Aktualisierung)

I.1.32 IT-Sicherheit

B122 IT-Sicherheit

| | |
|---------------------------------|--|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B122 |
| Bezeichnung | IT-Sicherheit |
| Lehrveranstaltung(en) | B122a IT-Sicherheit |
| Verantwortliche(r) | Prof. Dr. Gerd Beuster |
| Zuordnung zum Curriculum | E-Commerce (Bachelor) IT-Ingenieurwesen (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Smart Technology (Bachelor) Wirtschaftsinformatik (Bachelor) |
| Verwendbarkeit | Das Modul setzt grundlegende Kenntnisse der Programmierung und des Aufbaus eines Computersystems sowie von Computernetzen voraus. Nach Abschluss des Moduls verfügen die Studierenden über weiterführende Kenntnisse auf den Gebieten Computernetze, Kryptographie und Programmierung. Dies ist insbesondere verwendbar für Tätigkeiten und weiterführende Veranstaltungen im Bereich IT-Sicherheit. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Die Studierenden besitzen grundlegende Kenntnisse der Programmierung und des Aufbaus eines Computersystems sowie von Computernetzen. |
| Dauer | 1 |

Lernziele

Die Studierenden erwerben die notwendigen Kenntnisse, um Software-Systeme und ihre betrieblichen Einsatzszenarien in Hinblick auf Ihre Sicherheit einschätzen zu können. Weiterhin sind die Studierenden nach Absolvierung des Moduls in der Lage, bei der Konzeption und Entwicklung von Software-Systemen und in ihrem Unternehmenseinsatz relevante Sicherheitsaspekte zu berücksichtigen. Die Studierenden erwerben Kenntnisse im Bereich IT-Sicherheit nicht nur bezogen auf einzelne Software-Systeme, sondern auch im Hinblick auf die IT-Infrastruktur. Die Studierenden verfügen über das Wissen der verschiedenen Bedrohungs-

und Angriffsarten. Sie kennen die jeweiligen Maßnahmen zur Abwehr der Bedrohungen, insbesondere bei vernetzten Anwendungen.

I.1.32.1 IT-Sicherheit

| | |
|---------------------------------|---|
| Lehrveranstaltung | IT-Sicherheit |
| Dozent(en) | Gerd Beuster |
| Hörtermin | 6 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung mit integrierter Übung/Workshop/Assigm. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Prüfungsform | Klausur + ggf. Bonus |
| Sprache | deutsch/englisch |
| Lehr- und Medienform(en) | E-Learning |

Lernziele

Die Studierenden erwerben die notwendigen Kenntnisse, um Softwaresysteme und ihre betrieblichen Einsatzszenarien in Hinblick auf ihre Sicherheit einschätzen zu können. Sie sind in der Lage, bei der Konzeption und Entwicklung von Softwaresystemen und in ihrem Unternehmens-einsatz relevante Sicherheitsaspekte zu berücksichtigen.

- Kenntnis der unterschiedlichen Bedrohungsszenarien und -arten.
- Kenntnis der besonderen Gefahren bei internetbasierten Anwendungen.
- Kenntnis typischer primärer Sicherheitsziele (Vertraulichkeit, Authentifizierung, Verbindlichkeit, u.a.).
- Kenntnis der Verfahren zur Gewährleistung der unterschiedlichen Sicherheitsziele.
- Kenntnis der praxisrelevanten kryptografischen Verfahren und Protokolle.
- Kenntnis der Sicherungsmaßnahmen in Rechnernetzen.
- Fähigkeit, grundlegende Sicherungsmaßnahmen für Web-Anwendungen umzusetzen.
- Kenntnis der Bestandteile einer IT-Sicherheitsinfrastruktur und ihrer zentralen Funktionalitäten.
- Kenntnis der Verfahren zur Risikoabschätzung und Bewertung der Sicherheit von IT-Systemen und die Fähigkeit, diese anzuwenden.

Inhalt

- Gegenstandsbereich der IT-Sicherheit
- Aktuelle Richtlinien, Standards, Normen und Gesetze
- Bedrohungen der IT-Sicherheit und daraus resultierende Risiken
- Primäre Sicherheitsziele
- Überblick über Verfahren zur Erreichung der Ziele
- Kryptografische Verfahren
 - Verschlüsselungsverfahren

- * Symmetrische Verschlüsselungsverfahren
 - * Asymmetrische Verschlüsselungsverfahren
- Hash-Funktionen
- Schlüsselmanagement
- Zertifikate
- Kryptografische Protokolle
 - * Digitale Signatur
 - * Zeitstempel
 - * SSL / TLS-Protokoll
- Authentifizierungsverfahren
- Übertragungssicherheit in Netzen
 - Sichere IP-Kommunikation
 - VPN-Technologien
- Sicherheitsarchitekturen und ihre Komponenten
 - Sicherheitsaspekte von Web-Servern
 - Firewall-Systeme
 - Intrusion Detection-Systeme
- Sicherheit von Web-Anwendungen
- Technisch / organisatorische Maßnahmen zur Erhöhung der IT-Sicherheit
- Risiko- und Sicherheitsmanagement

Literatur

- Anderson, Ross J.: Security Engineering : A Guide to Building Dependable Distributed Systems. 2. Auflage. Hoboken (NJ), USA: Wiley & Sons, 2008.
- BSI - Bundesamt für Sicherheit in der Informationstechnik (Hrsg.): Informationssicherheit und IT-Grundschutz : BSI-Standards 100-1, 100-2 und 100-3. 2. Auflage. Köln : Bundesanzeiger Verlag, 2008.
- Eckert, Claudia: IT-Sicherheit : Konzepte - Verfahren - Protokolle. 8. Auflage München : Oldenbourg, 2013.
- Ferguson, Niels; Schneier Bruce, Kohno; Tadayoshi: Cryptography Engineering : Design Principles and Practical Applications. Hoboken (NJ), USA: Wiley & Sons, 2010.
- Kersten, Heinrich; Klett, Gerhard: Der IT Security Manager. 3. Auflage. Wiesbaden: Springer Vieweg, 2013.
- Pfleeger, Charls P.;Pfleeger, Shari Lawrence: Security in Computing. 4. Auflage. München: Prentice Hall, 2012.

- Progutke, Werner: Basiswissen IT-Sicherheit : Das Wichtigste für den Schutz von Systemen & Daten. 3. Auflage. Herdecke: W3L-Verlag, 2013.
- Stallings, William: Computer Security : Principles and Practice. 2. Auflage. München: Pearson, 2012.
- Stallings, William: Cryptography and Network Security : Principles and Practice. 6. Auflage. München: Pearson, 2014.
- Swoboda, Joachim; Spitz, Stephan; Pramateftakis, Michael: Kryptographie und IT-Sicherheit : Grundlagen und Anwendungen. 2. Auflage Wiesbaden: Vieweg + Teubner Verlag, 2011.
- Witt, Bernhard Carsten: IT-Sicherheit kompakt und verständlich : Eine praxisorientierte Einführung. Wiesbaden: Vieweg & Sohn, 2006.

I.1.33 Soft Skills

B118 Soft Skills

| | |
|---------------------------------|--|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B118 |
| Bezeichnung | Soft Skills |
| Lehrveranstaltung(en) | B118a Assistenz B118b Communication Skills |
| Verantwortliche(r) | jeweiliger Dozent |
| Zuordnung zum Curriculum | Betriebswirtschaftslehre (Bachelor) Computer Games Technology (Bachelor) Data Science & Artificial Intelligence (Bachelor) E-Commerce (Bachelor) IT-Ingenieurwesen (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Technische Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) |
| Verwendbarkeit | Die Inhalte dieses Moduls können gewinnbringend in Projekten, der Bachelor-Thesis und im täglichen Berufsleben genutzt werden. Das Modul sollte in allen Studiengängen verwendet werden. |
| Semesterwochenstunden | 5 |
| ECTS | 5.0 |
| Voraussetzungen | Fachliche Inhalte der ersten 4 Studiensemester |
| Dauer | 1 |

Lernziele

Nach Abschluss des Moduls haben die Studierenden die Fähigkeit erworben, in Kooperation mit den Dozenten und Assistenten, ihr Wissen und ihre Erfahrungen aus früheren Veranstaltungen der Betriebswirtschaftslehre, Mathematik und Informatik an Studierende jüngerer Semester weiter zu geben. Mit zunehmender Dauer des Semesters verbinden die Studierenden Kenntnisse aus der Veranstaltung "Communication Skills" mit ihrer Assistenz Tätigkeit.

I.1.33.1 Assistenz

| | |
|---------------------------------|--|
| Lehrveranstaltung | Assistenz |
| Dozent(en) | verschiedene Dozenten |
| Hörtermin | 6 |
| Häufigkeit | jedes Semester |
| Lehrform | Assistenz |
| Semesterwochenstunden | 3 |
| ECTS | 3.0 |
| Prüfungsform | Schriftl. Ausarbeitung (ggf. mit Präsentation) |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Tutorien |

Lernziele

Die Studierenden entwickeln unter Anleitung eines Hochschullehrers die Fähigkeiten ...

- fachspezifische Aufgabenstellungen zu analysieren
- problemspezifische Lösungen zu konzipieren und
- als Ergebnis begründet zu präsentieren.

Inhalt

Im Rahmen der Assistenz werden die Studierenden von den Hochschullehrern mit konkreten (Teil)-Projekten betraut. Diese können ein weites Spektrum umfassen. So sind z.B. die Durchführung kleinerer empirischer Umfragen oder auch die eigenständige Recherche und Ausarbeitung spezieller Fachinhalte denkbar. Ebenso in Betracht kommen die Durchführung von Tutorien oder Übungen. Die Assistenz ist selbständig zu bearbeiten und kann die Abstimmung mit anderen Studierenden erfordern.

Literatur

keine

I.1.33.2 Communication Skills

| | |
|---------------------------------|--|
| Lehrveranstaltung | Communication Skills |
| Dozent(en) | Anna-Magdalena Kölzer |
| Hörtermin | 6 |
| Häufigkeit | jedes Semester |
| Lehrform | Workshop |
| Semesterwochenstunden | 2 |
| ECTS | 2.0 |
| Prüfungsform | Schriftl. Ausarbeitung (ggf. mit Präsentation) |
| Sprache | deutsch |
| Lehr- und Medienform(en) | studentische Arbeit am Rechner |

Lernziele

Die Studierenden verfügen nach dem Besuch der Lehrveranstaltung über folgende Kompetenzen:

- Besitz verbesserter persönlicher Soft Skills, wie sie für Studium oder Beruf erforderlich sind
- Sensibilität für menschliche Interaktionen und Betriebsprozesse
- Besitz erweiterter rhetorischer Fähigkeiten im Rahmen von Präsentationen, Vorträgen und Referaten sowie sozialer Kompetenz
- Kenntnis der Bedeutung von verbalen und nonverbalen Signalen für die eigene Kommunikation sowie die Fähigkeit, diese zu erkennen
- Fähigkeit zum angemessenen Verhalten bei Teamarbeit oder Projekten
- Fähigkeit zur Selbstdarstellung bei Bewerbungen, Interviews, Assessment-Centern.

Inhalt

Im Rahmen des Workshops werden die folgenden Inhalte behandelt:

- Selbstanalyse (Fragebogen) als Grundlage für Bewerbungen
- Rhetorik & Präsentation (Theorie und Praxis)
- Struktur und Aufbau von Bewerbungsunterlagen
- Bewerbungsprozess
- Interview (Theorie und Praxis)
- Assessment Center (Theorie)
- Persönlichkeitsfragebogen und Testverfahren (Intelligenz und Konzentration) (Praxis)
- Gruppenübungen (Praxis)

Literatur

- ARNOLD, Frank:
Management von den besten lernen.
München: Hans Hauser Verlag, 2010
- APPELMANN, Björn:
Führen mit emotionaler Intelligenz.
Bielefeld: Bertelsmann Verlag, 2009
- BIERKENBIEHL, Vera F.:
Rhetorik, Redetraining für jeden Anlass. Besser reden, verhandeln, diskutieren.
12. Aufl. München: Ariston Verlag, 2010
- BOLLES, Nelson:
Durchstarten zum Traumjob. Das ultimative Handbuch für Ein-, Um- und Aufsteiger.
2. Aufl. Frankfurt/New York: Campus Verlag, 2009
- DUDENREDAKTION mit HUTH, Siegfried A.:
Reden halten - leicht gemacht. Ein Ratgeber.
Mannheim/Leipzig: Dudenverlag, 2007
- GRÜNING; Carolin; MIELKE; Gregor:
Präsentieren und Überzeugen. Das Kienbaum Trainingskonzept.
Freiburg: Haufe-Lexware Verlag, 2004
- HERTEL, Anita von:
Professionelle Konfliktlösung. Führen mit Mediationskompetenz.
Handelsblatt, Bd., 6, Kompetent managen.
Frankfurt: Campus Verlag, 2009
- HESSE, Jürgen; SCHRADER, Hans Christian:
Assessment-Center für Hochschulabsolventen.
5. Auflage, Eichborn: Eichborn Verlag, 2009
- MENTZEL, Wolfgang; GROTZFELD, Svenja; HAUB, Christine:
Mitarbeitergespräche.
Freiburg: Haufe-Lexware Verlag, 2009
- MORITZ, Andr; RIMBACH, Felix:
Soft Skills für Young Professional. Alles was Sie für ihre Karriere wissen müssen.
2. Aufl. Offenbach: Gabal Verlag, 2008
- PERTL, Klaus N.:
Karrierefaktor Selbstmanagement. So erreichen Sie ihre Ziele.
Freiburg: Haufe-Verlag, 2005
- PORTNER, Jutta:
Besser verhandeln. Das Trainingsbuch.
Offenbach: Gabal Verlag, 2010
- PÜTTJER, Christian; SCHNIERDA, Uwe:
Assessment-Center. Training für Führungskräfte.
Frankfurt/New York: Campus Verlag, 2009

- PÜTTJER, Christian; SCHNIERDA, Uwe:
Das große Bewerbungshandbuch.
Frankfurt: Campus Verlag, 2010
- SCHULZ VON THUN, Friedemann; RUPPEL, Johannes; STRATMANN, Roswitha:
Miteinander Reden. Kommunikationspsychologie für Führungskräfte.
10. Auflage, Reinbek bei Hamburg: rororo, 2003

I.1.34 Software-Projekt

B121 Software-Projekt

| | |
|---------------------------------|--|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B121 |
| Bezeichnung | Software-Projekt |
| Lehrveranstaltung(en) | B121b Softwareprojekt B121a Projektmanagement |
| Verantwortliche(r) | jeweiliger Dozent |
| Zuordnung zum Curriculum | Informatik (Bachelor) Medieninformatik (Bachelor) Wirtschaftsinformatik (Bachelor) |
| Verwendbarkeit | Dieses Modul dient als Vorbereitung für die praktischen Anteile einer Bachelor-Thesis. |
| Semesterwochenstunden | 6 |
| ECTS | 10.0 |
| Voraussetzungen | Voraussetzungen für dieses Modul sind Kenntnisse aus den Veranstaltungen Algorithmen und Datenstrukturen und Fortgeschrittener Objektorientierter Programmierung. Weiter werden Techniken zur Modellierung von Software (Entwurfsmuster, Abstrakte Syntax) vorausgesetzt. Elementare Fähigkeit zur Teamarbeit und Selbstorganisation sind ebenfalls notwendig. |
| Dauer | 1 |

Lernziele

Ziel dieses Projekts ist es, den Einsatz von von OO-Techniken in einer größeren Aufgabe in einem Team zu erfahren. Dabei ist die Themenstellung so gewählt dass zwischen den verschiedenen 2-er-Gruppen die Notwendigkeit der Koordination, Abstimmung und Diskussion, insbesondere über die Schnittstellen der Teilaufgaben, besteht, so dass die Bedeutung der sauberen Entwicklung von Schnittstellen erkannt wird.

Weiter wird die Modellierung und der Entwurf eines Software-Systems an nichttrivialen praxisnahen Problemstellungen trainiert, und so das Wissen aus der Veranstaltung über Software-Design angewandt und gefestigt.

Die soziale Kompetenz, Teamfähigkeit und Eigenverantwortung wird durch selbständige Projektplanung und Projektorganisation einschließlich Aufgabenaufteilung, Zeitplanung und Aufwandsschätzung trainiert.

Die Teamfähigkeit und die Kommunikationsfähigkeit wird gestärkt. Durch die Verwendung von fertigen Teilsystemen, Bibliotheken und Fremdsoftware, und auch dem Einsatz von Sprachen, die nicht intensiv in den Programmiersprachvorlesungen behandelt werden (Ruby, Python, ...) wird das selbständiges Einarbeiten in neue Umgebungen und Systeme trainiert und gefördert

In der Veranstaltung Projektmanagement ist das Ziel das Kennenlernen der grundlegenden Begriffe und Techniken entsprechend den Phasen eines Projekts und die Vermittlung grundlegender Kommunikationsfähigkeiten (präsentieren, diskutieren, moderieren und verhandeln).

Als Software-technischer Aspekt steht das Arbeiten im Team und das praktische Anwenden der Techniken des Programmieren im Großen im Mittelpunkt.

I.1.34.1 Softwareprojekt

| | |
|---------------------------------|-------------------------------|
| Lehrveranstaltung | Softwareprojekt |
| Dozent(en) | Christian-Arved Bohn |
| Hörtermin | 6 |
| Häufigkeit | jedes Semester |
| Lehrform | Projekt |
| Semesterwochenstunden | 4 |
| ECTS | 8.0 |
| Prüfungsform | Praktikumsbericht / Protokoll |
| Sprache | deutsch |
| Lehr- und Medienform(en) | |

Lernziele

Ziel dieses Projekts ist es, den Einsatz von von OO-Techniken in einer größeren Aufgabe in einem Team zu üben. Dabei ist die Themenstellung so gewählt dass zwischen den verschiedenen 2-er-Gruppen die Notwendigkeit der Koordination, Abstimmung und Diskussion, insbesondere über die Schnittstellen der Teilaufgaben, besteht.

Weiter sollen die Modellierung und der Entwurf eines Software-Systems an nichttrivialen praxisnahen Problemstellungen trainiert werden, und so das Wissen aus der Veranstaltung über Software-Design angewandt und gefestigt werden.

Die soziale Kompetenz, Teamfähigkeit und Eigenverantwortung wird durch selbständige Projektplanung und Projektorganisation einschließlich Aufgabenaufteilung, Zeitplanung und Aufwandsschätzung trainiert. Die Teamfähigkeit und die Kommunikationsfähigkeit werden gestärkt. Durch die Verwendung von fertigen Teilsystemen, Bibliotheken und Fremdsoftware, und auch dem Einsatz von Sprachen, die nicht intensiv in den Programmiersprachvorlesungen behandelt werden (Ruby, Python, ...) wird das selbständige Einarbeiten in neue Umgebungen und Systeme trainiert und gefördert.

Inhalt

Variierende Themen für Software-Projekte, unter anderem aus dem Web-Bereich, die mit objektorientierten Techniken zu lösen sind.

Literatur

Uwe Schmidt: Software-Projekt: Organisation und Themen, Unterlagen im Web:<http://www.fh-wedel.de/~si/praktika/SoftwarePraktikum/index.html>

I.1.34.2 Projektmanagement

| | |
|---------------------------------|-------------------|
| Lehrveranstaltung | Projektmanagement |
| Dozent(en) | Gerrit Remané |
| Hörtermin | 6 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 2 |
| ECTS | 2.0 |
| Prüfungsform | Klausur |
| Sprache | None |
| Lehr- und Medienform(en) | |

Lernziele

- Sie verstehen die spezifischen Charakteristika und Herausforderungen von Projekten (z.B. im Unterschied zu Prozessen)
- Sie können die wichtigsten Projektmanagement-Tools je Projektphase anwenden (Initiierung, Planung, Durchführung, Abschluss)
- Sie können wesentliche Konzepte und Methoden anwenden, um Mensch-bezogene Herausforderungen im Projektumfeld zu analysieren und zu lösen (z.B. Motivation, Feedback, Veränderung)
- Sie können geeignete Projektmanagement-Ansätze (Wasserfall vs. Agil) je nach Projekttyp auswählen

Inhalt

Im Rahmen der digitalen Transformation werden wiederkehrende Aufgaben zunehmend automatisiert. Einmalige Tätigkeiten hingegen lassen sich schwierig automatisieren und werden daher in Zukunft weiter an Bedeutung gewinnen. Diese einmaligen, temporären Aufgaben sind per Definition Projekte; nicht zuletzt aufgrund dieses Umstandes wird Projektmanagement eine der wichtigsten Fähigkeiten für eine erfolgreiche Karriere im 21. Jahrhundert.

Wie schwierig Projektmanagement in der Praxis ist, wird beispielsweise dadurch ersichtlich, dass mehr als 2 von 3 IT-Projekten ihre Ziele verfehlen. Auch wenn die Gründe hierfür im Einzelfall sehr unterschiedlich sein mögen, lassen diese sich doch in zwei breite Gruppen unterteilen. Zur ersten Gruppe zählen fehlende Projektmanagement-Kompetenzen wie Auswahl der Projektmethodik, Projektplanung oder Risikokontrolle. Zur zweiten Gruppe zählen ungenügende Soft Skills, um alle beteiligten Stakeholder zu managen, wie beispielsweise Motivation, Konfliktlösung oder Veränderungsmanagement.

Zielsetzung dieser Veranstaltung ist die Entwicklung wesentlicher Grundlagen in beiden Bereichen: Grundlegende Projektmanagementfähigkeiten (im engeren Sinne) sowie notwendige Softskills eines Projektleiters.

Kurzgliederung:

- Einführung in Projektmanagement
- Projektphasen (Initiierung, Planung, Durchführung, Abschluss)

- Soft Skills (Motivation, Veränderungsmanagement, Feedback, ...)
- Spezifische Ansätze (Wasserfall, Agil, Großprojekte, ...)

Literatur

- Verzuh: The Fast Forward MBA in Project Management, Fifth Edition, New Jersey, 2016
- Wysocki: Effective Project Management – Traditional, Agile, Extreme, Seventh Edition, Indianapolis, 2014
- PMI: A Guide to the Project Management Body of Knowledge (Pmbok Guide), Sixth Edition, Newton Square, 2017

I.1.35 Operations Research

B082 Operations Research

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B082 |
| Bezeichnung | Operations Research |
| Lehrveranstaltung(en) | B082a Operations Research B082b Übg. Operations Research |
| Verantwortliche(r) | Prof. Dr. Sebastian Iwanowski |
| Zuordnung zum Curriculum | Betriebswirtschaftslehre (Bachelor) E-Commerce (Bachelor) IT-Ingenieurwesen (Bachelor) Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) Wirtschaftsingenieurwesen (Bachelor) |
| Verwendbarkeit | Die Kenntnisse aus diesem Modul finden dort Anwendung, wo Optimierungsprobleme mathematisch gelöst werden. |
| Semesterwochenstunden | 5 |
| ECTS | 5.0 |
| Voraussetzungen | Grundlegende Kenntnisse der linearen Algebra. |
| Dauer | 1 |

Lernziele

Wichtigstes Lernziel des Moduls ist die Weiterentwicklung des Abstraktionsvermögens der Studierenden. Nach Abschluss des Moduls sind sie in der Lage, Problemstellungen als Operations Research-Aufgaben zu erkennen, aus den Problemstellungen mathematische Modelle abzuleiten und diese in Standardformen zu transformieren. Sie können die gelernten Lösungsverfahren der linearen Optimierung nutzen. Desweiteren verfügen sie über die Fähigkeit, die errechneten Ergebnisse als Lösungen für die gegebene Problemstellung zu interpretieren und kritisch zu bewerten.

I.1.35.1 Operations Research

| | |
|---------------------------------|---|
| Lehrveranstaltung | Operations Research |
| Dozent(en) | Sebastian Iwanowski |
| Hörtermin | 6 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 4 |
| ECTS | 4.0 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, Handout, interaktive Entwicklung und Diskussion von Modellen, Overheadfolien, Software-demonstration, Tafel |

Lernziele

Nach Abschluss der Veranstaltung besitzen die Studierenden folgende Kompetenzen:

- Kenntnis der mathematischen Methoden des Operations Research.
- Fähigkeit, Problemstellungen als Operations Research-Aufgaben zu erkennen, mathematische Modelle zu entwickeln und diese so in standardisierte Modelle zu transformieren, dass die gelernten Lösungsverfahren angewandt werden können.
- Fähigkeit, im Team komplexe Optimierungsprobleme anhand von Problemstellungen aus der Wirtschaft zu analysieren und dafür die mathematischen Modelle so zu entwickeln und zu transformieren, dass sie unter Verwendung eines Softwaresystems gelöst werden können.
- Fähigkeit zur Ergebnisinterpretation.
- Fähigkeit, sowohl beim Entwurf von Anwendungssystemen Methoden des Operations Research in dieselben zu integrieren als auch die Ergebnisse des Einsatzes von Operations Research-Methoden in diese Systeme als Grundlage für betriebliche Entscheidungsprozesse zu verwenden.

Inhalt

- Einführung in die lineare Optimierung
- Der Simplex-Algorithmus
- Sensitivitätsanalyse
- Das duale Problem
- Ganzzahlige lineare Optimierung
- Das Transportproblem
- Das Zuordnungsproblem
- Zielprogrammierung

Literatur

- Domschke, Wolfgang; Drexl, Andreas; Klein, Robert; Scholl, Armin; Voß, Stefan: Übungen und Fallbeispiele zum Operations Research. 7. Auflage. Springer: Berlin, 2011.
- Ellinger, Theodor; Beuermann, Günter; Leisten, Rainer: Operations Research : Eine Einführung. 6. Auflage Berlin Heidelberg: Springer-Verlag, 2003.
- Runzheimer, Bodo: Operations Research 1 : Lineare Planungsrechnung und Netzplantechnik. 8. Auflage. Wiesbaden: Springer Gabler, 2005.
- Suhl, Leena; Mellouli, Taieb: Optimierungssysteme : Modelle, Verfahren, Software, Anwendungen. 3. Auflage. Wiesbaden: Springer Gabler, 2013.
- Taha, Hamdy A.: Operations Research : An Introduction. 9. Auflage. München: Pearson, 2010.
- Werners, Brigitte: Grundlagen des Operations Research : Mit Aufgaben und Lösungen. 3. Auflage. Wiesbaden: Springer Gabler, 2013.
- Winston, Wayne L.: Operations Research : Applications and Algorithms. 4. Auflage. Boston (MA), USA: Cengage Learning Emea, 2003.
- Zimmermann, Werner; Stache, Ulrich: Operations Research : Quantitative Methoden zur Entscheidungsvorbereitung, 10. Auflage. Oldenbourg: Oldenbourg-Verlag, 2001.

I.1.35.2 Übg. Operations Research

| | |
|---------------------------------|--------------------------------|
| Lehrveranstaltung | Übg. Operations Research |
| Dozent(en) | Emre Kilic |
| Hörtermin | 6 |
| Häufigkeit | jedes Semester |
| Lehrform | Übung/Praktikum/Planspiel |
| Semesterwochenstunden | 1 |
| ECTS | 1.0 |
| Prüfungsform | Abnahme |
| Sprache | deutsch |
| Lehr- und Medienform(en) | studentische Arbeit am Rechner |

Lernziele

Studierende ...

- leiten lineare Programme aus informell umschriebenen praktischen Optimierungsproblemen ab.
- wenden die Simplexmethode zur eigenständigen Lösung linearer Programme an.
- nutzen Softwaretools zur Lösung linearer Optimierungsprobleme.
- interpretieren selbständig Lösungstableaus in Hinblick auf ökonomische Planungsprobleme.

Inhalt

- Lösung eines linearen Optimierungsproblems mittels Simplexmethode
 - Aufstellen des mathematischen Modells und Ableitung der kanonischen Form des primalen Problems
 - Lösung des primalen Problems
 - Aufstellen des mathematischen Modells und Ableitung der kanonischen Form des dualen Problems
 - Lösung des dualen Problems
- Ergebnis-Interpretation
 - Angabe und Interpretation der Lösung (Basisvariable, Nichtbasisvariable)
 - Interpretation von Schattenpreisen
 - Einordnung in den ökonomischen Kontext
 - Abbildung zwischen dualer und primaler Lösung
 - Erkennen und Interpretieren von Entartung / Mehrdeutigkeit
- Lösung eines linearen Optimierungsproblems mittels Excel-Solver
- Mündliche Abnahme der Ergebnisse

Literatur

siehe Vorlesung Operations Research

I.1.36 Grundlagen der Computergrafik

B085 Grundlagen der Computergrafik

| | |
|---------------------------------|--|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B085 |
| Bezeichnung | Grundlagen der Computergrafik |
| Lehrveranstaltung(en) | B085a Grundlagen der Computergrafik B085b Prakt. Grundlagen der Computergrafik |
| Verantwortliche(r) | Prof. Dr. Christian-Arved Bohn |
| Zuordnung zum Curriculum | Computer Games Technology (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Smart Technology (Bachelor) Technische Informatik (Bachelor) |
| Verwendbarkeit | Die generative 3D-Computergrafik berührt inhaltlich einige Grundgedanken der 2D Bildbearbeitung. Das Modul "Bildbearbeitung und -analyse" bietet sich daher als Kombination mit diesem Modul an. |
| Semesterwochenstunden | 6 |
| ECTS | 5.0 |
| Voraussetzungen | Grundkenntnisse der Mathematik, insbesondere der linearen Algebra und Vektorrechnung |
| Dauer | 1 |

Lernziele

Lernziele des Moduls sind die erlernten theoretischen Kompetenzen der Vorlesung "Grundlagen der Computergrafik" und die Fähigkeit, diese auch praktisch einsetzen zu können, was im zugehörigen Praktikum vermittelt wird.

I.1.36.1 Grundlagen der Computergrafik

| | |
|---------------------------------|---|
| Lehrveranstaltung | Grundlagen der Computergrafik |
| Dozent(en) | Christian-Arved Bohn |
| Hörtermin | 6 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 2 |
| ECTS | 2.0 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, Handout, Softwaredemonstration, Tafel |

Lernziele

Studierenden werden Fähigkeiten,

- grundlegende Probleme der generativen Computergrafik einzuordnen und zu klassifizieren und
- entsprechende Lösungsstrategien vorzuschlagen und zu implementieren,

vermittelt.

Inhalt

Die Veranstaltung gibt einen Überblick über wesentliche Aspekte der generativen Computergrafik. Konkret werden die Algorithmen des (a) Raytracings und der Projektion als geometrische Abbildung behandelt und jeweils die Teilaspekte Projektion, Verdeckung und Beleuchtungsrechnung dargestellt. Es werden intensiv Methoden der linearen Algebra besprochen, die geometrische Abbildungen zur Konstruktion von virtuellen Szenen und zur Projektion verwenden. Es folgen praktische Aspekte, die es hier zu beachten gilt und deren hardwarenahe Realisierung (z.B. "Clipping", "Buffer"). Die Technik der Texturierung wird aus mathematischer Sicht behandelt und anhand von praktischen Beispielen erläutert. Einen Einblick in weiterführende Probleme der Computergrafik geben die Grundlagen der globalen Beleuchtungsrechnung ("Rendering Equation").

Literatur

- Hearn, M. P. Baker: Computer Graphics with OpenGL, Pearson Education International, 2004.
- Peter Shirley et al.: Fundamentals of Computer Graphics, A K Peters, 2005.

I.1.36.2 Prakt. Grundlagen der Computergrafik

| | |
|---------------------------------|---|
| Lehrveranstaltung | Prakt. Grundlagen der Computergrafik |
| Dozent(en) | Lars Neumann |
| Hörtermin | 6 |
| Häufigkeit | jährlich |
| Lehrform | Übung/Praktikum/Planspiel |
| Semesterwochenstunden | 4 |
| ECTS | 3.0 |
| Prüfungsform | Abnahme |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, Handout, Softwaredemonstration, studentische Arbeit am Rechner, Tafel, Tutorien |

Lernziele

Die Studierenden

- sind in der Lage, die in der Vorlesung vermittelten theoretischen Hintergründe selbst praktisch mit OpenGL anzuwenden und sicher zu handhaben,
- haben umfangreiche Kenntnisse und praktische Erfahrungen zu den Themen OpenGL, Callback-Prinzip, 2D-Anwendungen, Matrixstacks, 3D-Szenen, Displaylisten, lokale Beleuchtung, Texturierung, Picking, Viewports und Blending,
- haben sich die Grundlagen von Vertex-Arrays, Simulationen und Shadern erarbeitet und
- steigern ihre Teamfähigkeit durch intensive Arbeit in Zweierteams und Kommunikation über auftretende Probleme in der ganzen Gruppe.

Inhalt

OpenGL, affine Transformationen, 2D- und 3D-Anwendungen, lokale Beleuchtungsmodelle, Texturierung, Picking, Viewports, Transparenz, Vertex-Arrays, Simulationen, Shader

Literatur

Skript:

- Vorlesungsskript unter <http://cg.viswiz.de/> => Lehrveranstaltungen => Computergrafik 1
- Weiteres Material unter <http://www.fh-wedel.de/mitarbeiter/ne/praktikum-grundlagen-der-computergrafik-fh/material/>

Online-Quellen:

- The OpenGL Programming Guide - The Redbook (<http://www.glprogramming.com/red/>)
- The OpenGL Reference Manual - The Bluebook (<http://www.glprogramming.com/blue/>)
- NeHe Productions (<http://nehe.gamedev.net/>)

Bücher:

- Computergrafik und OpenGL - Eine systematische Einführung, Dieter Orlamünder / Wilfried Mascolus, Hanser, 2004, ISBN: 3-446-22837-3
- Jetzt lerne ich OpenGL : der einfache Einstieg in die Schnittstellenprogrammierung, Lorenz Burggraf, Markt und Technik, 2003, ISBN: 3-8272-6237-2

I.1.37 Echtzeitsysteme

B101 Echtzeitsysteme

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B101 |
| Bezeichnung | Echtzeitsysteme |
| Lehrveranstaltung(en) | B101a Echtzeitsysteme B101b Prakt. Echtzeitsysteme B101a Interface-Technologie |
| Verantwortliche(r) | Prof. Dr. Sergei Sawitzki |
| Zuordnung zum Curriculum | IT-Ingenieurwesen (Bachelor) Informatik (Bachelor) Smart Technology (Bachelor) Technische Informatik (Bachelor) |
| Verwendbarkeit | <p>Das Modul "Echtzeitsysteme" baut auf den in den Modulen "Algorithmen und Datenstrukturen" und (im geringeren Maße) "Systemnahe Programmierung" erworbenen Kenntnissen und Fähigkeiten auf und vertieft diese im Bezug auf Echtzeitanwendungen. Das Modul kann sinnvoll durch die Module, die allgemeine Aspekte von Betriebssystemen (ohne besonderen Augenmerk auf Echtzeitfähigkeit) vermitteln, ergänzt werden. Auch eine Vertiefung durch die Module, die aktuelle industrielle Standards und Anwendungen betrachten, ist denkbar. Das Modul ist fachübergreifend in allen informatik-affinen Studiengängen einsetzbar, die Kompetenzen zum Echtzeit-Betrieb von Rechnersystemen vermitteln. Das Modul hat einen direkten Bezug zum Studiengangsziel des Studiengangs "Bachelor Technische Informatik" softwarenahe Kompetenzen in technischen Anwendungen der Informatik zu erwerben.</p> |
| Semesterwochenstunden | 6 |
| ECTS | 5.0 |
| Voraussetzungen | Es werden Kenntnisse der Programmiergrundlagen (Datentypen, Programmstrukturen), Basiskenntnisse aus der Informations- und Digitaltechnik (Zahlendarstellungen, Ablauf der Befehlsausführung, Daten- und Steuerfluss) sowie Beherrschung der grundlegenden Methoden und Verfahren der linearen Algebra (Matrixoperationen, Lösung von linearen Gleichungssystemen) vorausgesetzt. |

Lernziele

Das Modul vermittelt Kenntnisse und Fertigkeiten, die für die Konzipierung, Entwurf, Inbetriebnahme sowie den Umgang mit Echtzeitsystemen benötigt werden. Dabei werden gleichermaßen Software- und Hardware-Konzepte betrachtet. Einerseits werden Methoden und Mechanismen vorgestellt, mit denen Systeme von nebenläufigen, kooperierenden oder konkurrierenden Prozessen modelliert und implementiert werden. Andererseits werden hardwaretechnische Voraussetzungen für Echtzeit-Betrieb diskutiert, sowie die Unterschiede, die ein Echtzeit-Betriebssystem im Vergleich mit gewöhnlichem Betriebssystem aufweist. Schließlich werden relevante Aspekte der Ereigniserfassung und -verarbeitung im einem Rechnersystem besprochen, untermauert durch Einsatzbeispiele aus der industriellen Praxis. Auch Sicherheits- und Zuverlässigkeitsaspekte werden betrachtet. Durch selbständiges Lösen einer praktischen Aufgabenstellung aus dem Bereich Echtzeitsysteme sollen die Studierenden ihren Lernerfolg überprüfen.

I.1.37.1 Echtzeitsysteme

| | |
|---------------------------------|---|
| Lehrveranstaltung | Echtzeitsysteme |
| Dozent(en) | Sergei Sawitzki |
| Hörtermin | 6 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 2 |
| ECTS | 1.5 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Handout, Overheadfolien, Softwaredemonstration, Tafel |

Lernziele

Die Studierenden ...

- bekommen die Kenntnisse vermittelt, die für die systemnahe Programmierung benötigt werden; insbesondere diejenigen Softwarekenntnisse, die für die Programmierung von Embedded Systems, Systemen mit kleinen Hardwareressourcen und Echtzeitsystemen erforderlich sind.
- erlernen Modellierungstechniken und -methoden bei Programmierung von nebenläufigen Prozessen, Modellierung und Anwendung von Prozesskommunikations- und Synchronisationsmechanismen; insbesondere die Fähigkeit, Aufgabenstellungen auf Systeme nebenläufiger (kooperierender und konkurrierender) Prozesse abzubilden und diese softwaretechnisch umzusetzen.
- bekommen eine Einführung in die Methodik der Programmierung paralleler Prozesse, unter besonderer Berücksichtigung von Echtzeitanforderungen vermittelt und erwerben die Befähigung, einfache Echtzeitsysteme zu konzipieren und zu realisieren. Neben Kenntnis wesentlicher theoretischer Konzepte wird besonders auch Fähigkeit zu deren praktischer Umsetzung, die im Projekt Echtzeitsysteme durchgeführt wird, vermittelt.
- weisen durch Entwurf und Realisierung eines Echtzeit-Multitasking-Programms die Kompetenz nach, derartige Systeme aus einem Anforderungskatalog zu entwerfen und praktisch umzusetzen.
- trainieren die Arbeitskoordination, Teamfähigkeit und soziale Kompetenz durch die im Projekt praktizierte Teamarbeit.

Inhalt

- Einleitung
 - Lernziele
 - Organisatorisches
 - Literatur
 - Bezeichnungen und Konventionen
 - Definition und Einordnung

- Historische Entwicklung
- Prozesse
 - Grundbegriffe
 - Technische Umsetzung
 - Aufgaben des Betriebssystems
 - Probleme und Lösungsansätze
- Kommunikationsmechanismen
 - Übersicht
 - Semaphore
 - Monitore
 - Mailbox-Kommunikation
 - Nachrichtenaustausch
 - Weitere Mechanismen
 - Äquivalenzen und Beispiele
- Modellierung
 - Einleitung
 - Flussdiagramme
 - Petri-Netze
 - Weitere Modellierungstechniken
- Scheduling
 - Einleitung
 - Strategien
 - Zeitverwaltung
 - Beispiele

Literatur

- Tanenbaum, Andrew: Moderne Betriebssysteme, Pearson Studium, 3. Auflage, 2009
- Witzak, Michael: Echtzeit Betriebssysteme, Franzis Verlag, 2000
- Baumgarten, Bernd: Petri-Netze, Wissenschaftsverlag, 1990
- Labrosse, Jean: MicroC/OS-II, CMP Books, 2002
- Quade, Jürgen; Mächtel, Michael: Moderne Realzeitsysteme kompakt, dpunkt.verlag, 2012

I.1.37.2 Prakt. Echtzeitsysteme

| | |
|---------------------------------|---|
| Lehrveranstaltung | Prakt. Echtzeitsysteme |
| Dozent(en) | Timm Bostelmann |
| Hörtermin | 6 |
| Häufigkeit | jährlich |
| Lehrform | Übung/Praktikum/Planspiel |
| Semesterwochenstunden | 2 |
| ECTS | 2.0 |
| Prüfungsform | Abnahme |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Beamerpräsentation, Handout, Softwaredemonstration, studentische Arbeit am Rechner, Tafel |

Lernziele

Nachdem Studierende die Veranstaltung besucht haben, können sie

- ausgehend von einer Aufgabenstellung in Form eines Anforderungskatalogs ein System nebenläufiger (kooperierender und konkurrierender) Prozesse entwerfen.
- ein solches System unter Verwendung eines Echtzeitbetriebssystems praktisch umsetzen.
- sich mit Hilfe eines Handbuches in die Schnittstelle eines Echtzeitbetriebssystems einarbeiten.
- ihre Lösung auf System- und Implementierungsebene in einer schriftlichen Ausarbeitung dokumentieren.

Inhalt

- Einführungsvorlesung
 - Motivation zur Veranstaltung
 - Beschreibung der Systemumgebung
 - Einführung in die Verwendung des Echtzeitkerns
- Einführungsaufgabe
 - Geführtes Erstellen eines Beispielprojektes
 - Teilweise geführte Programmierung eines Prozesses
 - Selbstständige Programmierung eines Prozesses
- Eine Aufgabe aus dem Umfeld Echtzeit, Multitasking, Simulation wird gestellt und steht den Studierenden als Anforderungskatalog zur Verfügung
 - Struktureller Programmentwurf
 - Kodierung und Test
 - Erstellung einer Dokumentation
 - Abnahme durch den Betreuer

Literatur

Labrosse, Jean: MicroC/OS-II, CMP Books, 2002

I.1.37.3 Interface-Technologie

| | |
|---------------------------------|-----------------------|
| Lehrveranstaltung | Interface-Technologie |
| Dozent(en) | Dennis Säring |
| Hörtermin | 6 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung |
| Semesterwochenstunden | 2 |
| ECTS | 1.5 |
| Prüfungsform | Klausur |
| Sprache | deutsch |
| Lehr- und Medienform(en) | E-Learning |

Lernziele

Die Studierenden ...

- sind vertraut mit den grundlegenden Begriffen der Prozessdatenverarbeitung
- besitzen ein Verständnis für die sensorgestützte Aufnahme und Verarbeitung von Echtzeitdaten
- kennen die mathematischen Modelle für eine prädiktive Zustandsschätzungen auf Basis zurückliegender Messergebnisse
- können die Anforderungen an moderne BUS-Systeme im Kontext von komplexen Kommunikationsstrukturen (z.B. PKW, Flugzeug) einschätzen

Inhalt

- Grundlagen der Prozessdatenverarbeitung (PDV)
- Sensortechniken
- Strategien zur Fusionierung von Prozessdaten
- Prädiktionsmodell Kalman- und Partikel-Filter
- Moderne BUS-Systeme

Literatur

- Börcsök: Prozeßrechner- und Automation, Heise-Verlag, 1997
- Jacobsen: Einführung in die Prozeßdatenverarbeitung, Hanser-Verlag, 1996
- Wittgruber: Digitale Schnittstellen und BUS-Systeme, Vieweg-Verlag, 1999

I.1.38 Applied Data Science and Machine Learning

B209 Applied Data Science and Machine Learning

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B209 |
| Bezeichnung | Applied Data Science and Machine Learning |
| Lehrveranstaltung(en) | B209a Applied Data Science and Machine Learning |
| Verantwortliche(r) | Prof. Dr. Ulrich Hoffmann |
| Zuordnung zum Curriculum | Computer Games Technology (Bachelor) Data Science & Artificial Intelligence (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Smart Technology (Bachelor) |
| Verwendbarkeit | Das Modul lässt sich sinnvoll mit dem Modulen "Bildverarbeitung- und Analyse" und "Statistik" kombinieren, bei denen Grundlagen und ein wesentliches Anwendungsgebiet des maschinellen Lernens vermittelt werden. In den Modulen "Projekt Intelligente Systeme" und "Projekt Intelligente Umgebungen" können die erworbenen Kompetenzen in umfangreichen, interdisziplinären Projekten je nach gewählter Aufgabe eingesetzt werden. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Voraussetzungen | Sie Studierenden sollten solide mathematische Grundkenntnisse besitzen, wie sie in den Modulen "Analysis" und "Deskriptive Statistik & Grundlagen der Linearen Algebra" vermittelt werden. Sie sollten zudem über Fähigkeiten in der imperativen und objekt-orientierten Programmierung verfügen ("Programmstrukturen 1", "Programmstrukturen 2"). |
| Dauer | 1 |

Lernziele

Die Studierenden besitzen Kenntnisse über wesentliche Fragestellungen des maschinellen Lernens. Sie kennen das systematische Vorgehen bei der Durchführung von Machine-Learning-Projekten, sind mit den dabei beteiligten Schritten vertraut und kennen die dabei auftretenden Herausforderungen. Sie kennen wesentliche Begriffe des Maschinellen Lernens und können sie gezielt und präzise einsetzen. Sie kennen die unterschiedlichen Machine-Learning-Aufgaben,

verschiedene Machine-Learning-Verfahren (Algorithmen) und können ihr Einsatzgebiet, ihre Arbeitsweise und Eigenschaften benennen. Sie können Machine-Learning-Projekte bezüglich Ihrer Qualität systematisch untersuchen und bewerten.

Durch die durchgeführten Projekte im praktischen Teil des Modules besitzen die Studierende Kenntnisse und Fähigkeiten im Umgang mit ausgewählten Machine-Learning-Werkzeugen. Sie besitzen Kenntnisse über verschiedene weitere Werkzeuge und ihre Anwendungsgebiete. Sie verfügen über die Fähigkeit gegebene Daten für das maschinelle Lernen aufzubereiten bzw. die Daten auf Eignung zum maschinellen Lernens zu überprüfen.

I.1.38.1 Applied Data Science and Machine Learning

| | |
|---------------------------------|--|
| Lehrveranstaltung | Applied Data Science and Machine Learning |
| Dozent(en) | Christo Zonnev |
| Hörtermin | 6 |
| Häufigkeit | jährlich |
| Lehrform | Vorlesung mit integrierter Übung/Workshop/Assigm. |
| Semesterwochenstunden | 4 |
| ECTS | 5.0 |
| Prüfungsform | Schriftl. Ausarbeitung (ggf. mit Präsentation) |
| Sprache | None |
| Lehr- und Medienform(en) | Beamerpräsentation, interaktive Entwicklung und Diskussion von Modellen, Online-Aufbereitung, Softwaredemonstration, studentische Arbeit am Rechner, Tafel |

Lernziele

Nach Abschluss des Moduls besitzen die Studierenden...

- Kenntnisse über wesentliche Fragestellungen des maschinellen Lernens
- Kenntnisse über das systematische Vorgehen bei der Durchführung von Machine-Learning-Projekte und die beteiligten Schritten
- Kenntnisse der auftretenden Herausforderungen im Machine-Learning-Projekten
- Kenntnisse wesentlicher Begriffe des Maschinellen Lernens und die Fähigkeit sie gezielt und präzise einzusetzen
- Kenntnisse unterschiedlicher Machine-Learning-Aufgaben, verschiedener Machine-Learning-Verfahren (Algorithmen)
- die Fähigkeit die Einsatzgebiete, Arbeitsweise und Eigenschaften von Machine-Learning-Verfahren zu benennen
- Fähigkeit Machine-Learning-Projekte bezüglich Ihrer Qualität systematisch zu untersuchen und zu bewerten.
- Kenntnisse über verschiedene Machine-Learning-Werkzeugen und ihre Anwendungsgebiete
- Fähigkeiten im Umgang mit ausgewählten Machine-Learning-Werkzeugen
- die Fähigkeit gegebene Daten für das maschinelle Lernen aufzubereiten
- die Fähigkeit Daten auf Eignung zum maschinellen Lernens zu überprüfen

Inhalt

Vorlesung

- Einführung
 - Maschinelles Lernen - warum, wie, was / Grundlagen

- Data-Science-Grundlagen
 - Wiederholung der erforderlichen Mathematik & Statistik
 - Daten-Algorithmen
 - Einführung in allgemeinen Software-Werkzeuge für Data-Science
- Grundlagen des maschinellen Lernen
 - Daten-Algorithmen
- Einführung in Software-Werkzeuge des maschinellen Lernens
- Graph-Daten
 - Verarbeitung von Graphenstrukturen
 - Soziale Netzwerke
- neuronale Netze
 - neuronale Netze im Detail
 - Software-Werkzeug Keras
 - Software-Werkzeug Tensorflow
- Kursretrospektive

Praktischer Teil

- Projektaufgeenstellung
 - Ideenfindung
 - Aufgabenverteilung
- Projekt-Präsentationen

Literatur

- Bishop: Neural Networks for Pattern Recognition, Oxford Press 1995
- Sutton, Barto: Reinforcement Learning: An Introduction, MIT Press, Cambridge, MA, 1998
- Brause: Neuronale Netze, Teubner, 1991
- Raschka: Python Machine Learning, Packt, 2015
- Müller, Guido: Introduction to Machine Learning with Python, O'Reilly, 2016
- Richert, Coelho: Building Machine Learning Systems with Python, Packt, 2018

- Goodfellow: Deep Learning (Adaptive Computation and Machine Learning)
MIT Press, 2017
- Géron: Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools,
and Techniques for Building Intelligent Systems
O'Reilly, 2017

I.1.39 Auslandssemester

B099 Auslandssemester

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B099 |
| Bezeichnung | Auslandssemester |
| Lehrveranstaltung(en) | B099a Auslandssemester |
| Verantwortliche(r) | Dipl.-Soz. (FH) Nicole Haß |
| Zuordnung zum Curriculum | Betriebswirtschaftslehre (Bachelor) Computer Games Technology (Bachelor) Data Science & Artificial Intelligence (Bachelor) E-Commerce (Bachelor) IT-Ingenieurwesen (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Smart Technology (Bachelor) Technische Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) Wirtschaftsingenieurwesen (Bachelor) |
| Verwendbarkeit | Studierende sammeln sprachliche Erfahrungen und erweitern ihre sozialen Kompetenzen, die sie in ihr Berufsleben nach Studiumsabschluss einbringen können. |
| Semesterwochenstunden | 25 |
| ECTS | 30.0 |
| Voraussetzungen | Für eine Zulassung müssen alle Übergangsleistungen gemäß § 16a der Prüfungsverfahrensordnung und insgesamt mindestens 45 ECTS-Punkte erfolgreich absolviert sein. |
| Dauer | 1 |

Lernziele

Die fachlichen Lernziele dieses Moduls werden von den ausländischen Hochschulen festgelegt. Die FH Wedel prüft, ob diese Ziele inhaltlich vergleichbar und anerkenbar sind mit den eigenen Zielen.

Im Bereich soziale Kompetenz ist das Ziel das Kennenlernen einer anderen sprachlichen und kulturellen Umgebung und das Arbeiten und Kommunizieren in dieser. Außerdem natürlich das Erlernen und/oder Festigen einer Fremdsprache.

I.1.39.1 Auslandssemester

| | |
|---------------------------------|---|
| Lehrveranstaltung | Auslandssemester |
| Dozent(en) | Nicole Haß |
| Hörtermin | 6 |
| Häufigkeit | jedes Semester |
| Lehrform | Veranstaltungen an ausländischer Hochschule |
| Semesterwochenstunden | 25 |
| ECTS | 30.0 |
| Prüfungsform | Ausland |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Keine |

Lernziele

Nach Abschluss des Auslandssemester besitzen die Studierenden ...

- fundierte Sprachkompetenzen in englischer, französischer oder spanischer Sprache.
- erweiterte Kenntnisse über die Kultur des Gastlandes.

Inhalt

Verpflichtendes Auslandssemester:

Für ein verpflichtendes Auslandssemester muss der Umfang der erfolgreich zu erbringenden Leistungen (ohne Englisch-Sprachkurs) mindestens 30 ECTS-Punkte betragen oder einen entsprechenden gleichwertigen Umfang in lokalen Credits aufweisen. An der ausländischen Hochschule sind fachspezifische weiterführende und keine Grundlagenkurse zu belegen. Diese sollen im Zusammenhang mit dem Wedeler Studiengang stehen (hinsichtlich der zu belegenden Fächer gemäß Modulhandbuch).

Freiwilliges Auslandssemester:

Für ein freiwilliges Auslandssemester ist der Umfang der zu leistenden ECTS-Punkte (bzw. der gleichwertige Umfang in lokalen Credits) in der jeweiligen Studienordnung vorgegeben. An der ausländischen Hochschule sind fachspezifische Kurse zu belegen, die mit dem in Wedel belegten Studiengang in ergänzendem Zusammenhang stehen. Das Studienprogramm wird vor der Abreise individuell mit dem International Office vereinbart.

Literatur

Abhängig von der ausländischen Hochschule

I.1.40 Praxissemester (dual)

B176 Praxissemester (dual)

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B176 |
| Bezeichnung | Praxissemester (dual) |
| Lehrveranstaltung(en) | B176a Praxissemester (dual) |
| Verantwortliche(r) | Dipl.-Kauff. (FH) Journalistin Anke Amsel |
| Zuordnung zum Curriculum | Betriebswirtschaftslehre (Bachelor) Computer Games Technology (Bachelor) Data Science & Artificial Intelligence (Bachelor) E-Commerce (Bachelor) IT-Ingenieurwesen (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Smart Technology (Bachelor) Technische Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) Wirtschaftsingenieurwesen (Bachelor) |
| Verwendbarkeit | Verwendung der erworbenen Fähigkeiten in der späteren praxisorientierten Bachelor-Thesis. |
| Semesterwochenstunden | 20 |
| ECTS | 25.0 |
| Voraussetzungen | <p>Voraussetzung für die Zulassung zum Praxissemester ist der Nachweis der vorherigen studienbegleitenden Praxisblöcke. Sie dienen der Einarbeitung in die betriebliche Praxis. Die Praxisblöcke sind in Form von Berichten zu dokumentieren.</p> <p>Für eine Zulassung müssen alle Übergangsleistungen gemäß Studienordnung und insgesamt mindestens 75 ECTS-Punkte erfolgreich absolviert sein. Das Praxissemester darf nicht vor dem lt. Studienverlaufsplan festgelegten Semester angetreten werden. Es kann auf Antrag an den Prüfungsausschuss vorgezogen werden, wenn zu erwarten ist, dass die beziehungsweise der Studierende zum Zeitpunkt der Aufnahme des Praxissemesters die gemäß Studienordnung bis zum Praxissemester zu erwerbenden ECTS-Punkte erworben haben wird.</p> |

Die Anmeldung des Praxissemesters erfolgt bei der Koordinierungsstelle „Duale Studiengänge“ über ein Formblatt.

Dauer

1

Lernziele

Das Praxissemester bietet den Studierenden die Möglichkeit eine Verbindung von studien-gangsspezifischem und unternehmensspezifischem Kompetenzprofil herauszubilden.

Die Studierenden sollen im Kooperationsunternehmen in einer Vielzahl von Tätigkeitsfeldern qualifiziert an einem größeren Projekt mit Bezug zum Studiumsziel in eigener Verantwortung unter Anleitung erfahrener Mitarbeiter mitwirken. Die projektbezogene betriebliche Tätigkeit kann sich auf mehrere unabhängige Teilprojekte erstrecken.

Dabei sollen sich die Studierenden mit Leitfragen ihres Studiengangs auseinandersetzen. Die inhaltliche Vertiefung kann durch die Einbindung des Kooperationsunternehmens teilweise über das Lehrangebot der FH Wedel hinausgehen.

Durch das projektbezogene Arbeiten werden analytische, organisatorische, kommunikative und repräsentative Techniken eingeübt sowie die Verbindung zu Anwendungsgebieten des Studien-gangs hergestellt. Ziel ist der Theorietransfer in die jeweiligen betrieblichen Funktionsbereichen. Berufliche Realität soll erlebt und erlernt werden. Die Studierenden wählen wissenschaftliche Methoden, um Aufgaben des Berufslebens zu lösen.

Den Nachweis, dass sie ihr erlerntes Wissen auf eine anwendungsbezogene Aufgabenstellung aus einem Fachgebiet selbstständig auf wissenschaftlicher Grundlage im Rahmen des festgelegten Themas anwenden können, erbringen die Studierenden im Rahmen einer wissenschaftlichen Arbeit.

I.1.40.1 Praxissemester (dual)

| | |
|---------------------------------|-------------------------------|
| Lehrveranstaltung | Praxissemester (dual) |
| Dozent(en) | Anke Amsel |
| Hörtermin | 6 |
| Häufigkeit | jedes Semester |
| Lehrform | Praktikum |
| Semesterwochenstunden | 20 |
| ECTS | 25.0 |
| Prüfungsform | Praktikumsbericht / Protokoll |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Keine |

Lernziele

Die Studierenden ...

- können das Wissen aus dem bisherigen Studium in der Praxis anwenden und hinsichtlich der Tauglichkeit kritisch bewerten
- erweitern ihre wissenschaftlichen Ausbildung durch systematische praktische Erfahrungen
- können Projekten vorbereiten, analysieren und im Nachgang evaluieren
- bewerten Problemstellungen und können Lösungsansätze dafür entwickeln
- können Projektmanagement betreiben, Aktivitäten koordinieren, Planabweichungen hinterfragen.
- sehen und bewerten unternehmensweite und gesellschaftliche Zusammenhänge der eigenen Tätigkeit und zeigen ihre professionelle persönliche Qualifikation in der Zusammenarbeit mit Vorgesetzten und Kollegen
- nehmen Stellung zu den sozial-, datenschutz- oder umweltschutzbedingten Restriktionen bei der Umsetzung von betrieblichen Anforderungen
- reflektieren ihre Qualifikation und ihre eigene Tätigkeit
- sind zum selbständigen und eigenverantwortlichen Arbeiten fähig
- übernehmen Verantwortung für die Qualität der eigenständig übernommenen Arbeit
- entscheiden sich für systematische Vorgehensweisen und Arbeitstechniken
- klassifizieren ihre Tätigkeit zu Anwendungsgebieten des Studiengangs
- sind in der Lage das Wissen aus dem bisherigen Studium in der Praxis anzuwenden, fortzubilden und hinsichtlich der Tauglichkeit kritisch zu bewerten.

Inhalt

Der Inhalt des "Praxissemesters" muss mit der Zielrichtung des Studienganges vereinbar sein. Zur Erreichung dieses Ziel werden je Studiengang entsprechende Leitfragen formuliert. Von diesen sollen während des Praxissemester mindestens vier Themenkomplexen abgearbeitet und mindestens ein Themenkomplex vertieft werden.

Ausgehend von den Modulzielen des jeweiligen Studiengangs legt die/der Hochschulbetreuer in Absprache mit der/dem Studierenden fest, welche Themenkomplexe im Unternehmen bearbeitet werden sollen.

Literatur

themenabhängig

I.1.41 Wissenschaftliche Ausarbeitung (dual)

B179 Wissenschaftliche Ausarbeitung (dual)

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B179 |
| Bezeichnung | Wissenschaftliche Ausarbeitung (dual) |
| Lehrveranstaltung(en) | B179a Wissenschaftliche Ausarbeitung (dual) |
| Verantwortliche(r) | Dipl.-Kauff. (FH) Journalistin Anke Amsel |
| Zuordnung zum Curriculum | Betriebswirtschaftslehre (Bachelor) Computer Games Technology (Bachelor) Data Science & Artificial Intelligence (Bachelor) E-Commerce (Bachelor) IT-Ingenieurwesen (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Smart Technology (Bachelor) Technische Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) Wirtschaftsingenieurwesen (Bachelor) |
| Verwendbarkeit | Die wissenschaftliche Ausarbeitung dient als Vorbereitung auf den wissenschaftlichen Teil der Bachelor-Thesis. |
| Semesterwochenstunden | 3 |
| ECTS | 5.0 |
| Voraussetzungen | <p>Voraussetzung für die Zulassung im Rahmen des Praxissemesters ist der Nachweis der vorherigen studienbegleitenden Praxisblöcke. Sie dienen der Einarbeitung in die betriebliche Praxis. Die Praxisblöcke sind in Form von Berichten zu dokumentieren.</p> <p>Für eine Zulassung müssen alle Übergangsleistungen gemäß Studienordnung und insgesamt mindestens 75 ECTS-Punkte erfolgreich absolviert sein. Das Praxissemester darf nicht vor dem lt. Studienverlaufsplan festgelegten Semester angetreten werden. Es kann auf Antrag an den Prüfungsausschuss vorgezogen werden, wenn zu erwarten ist, dass die beziehungsweise der Studierende zum Zeitpunkt der Aufnahme des Praxissemesters die gemäß Studienordnung</p> |

bis zum Praxissemester zu erwerbenden ECTS-Punkte erworben haben wird.

Die Anmeldung erfolgt über ein Formblatt bei der beziehungsweise dem hochschulseitigen Betreuer(in).

Dauer

1

Lernziele

Die wissenschaftliche Ausarbeitung soll im Dualen Studium bestehende fachliche Lücken im Vergleich zum Vollzeitstudium ausgleichen und ggf. Bezüge zur betrieblichen Praxis aufweisen.

Es werden Kompetenzen zu Zitieren, Recherche, Verfassen wissenschaftlicher Texte und Methoden wissenschaftlichen Arbeitens erworben.

I.1.41.1 Wissenschaftliche Ausarbeitung (dual)

| | |
|---------------------------------|--|
| Lehrveranstaltung | Wissenschaftliche Ausarbeitung (dual) |
| Dozent(en) | jeweiliger Dozent |
| Hörtermin | 6 |
| Häufigkeit | jedes Semester |
| Lehrform | Praktikum |
| Semesterwochenstunden | 3 |
| ECTS | 5.0 |
| Prüfungsform | Schriftl. Ausarbeitung (ggf. mit Präsentation) |
| Sprache | deutsch |
| Lehr- und Medienform(en) | Keine |

Lernziele

Die Studierenden sind in der Lage, innerhalb einer vorgegebenen Frist ein Themenkomplex des Praxissemesters selbstständig wissenschaftlich zu bearbeiten und diesen kritisch zur praktischen Anwendung zu betrachten.

Inhalt

Die wissenschaftliche Arbeit ist als abschließende, vom Studierenden eigenständig aber hochschul- und unternehmensseitig betreute Ausarbeitung zum Praxissemester zu verstehen. Der Themenkomplex wird im Sinne der Zielsetzung des Praxissemesters mit der/dem hochschulseitigen Betreuer(in) abgestimmt und soll Bezüge zur betrieblichen Praxis aufweisen.

Literatur

themenabhängig

I.1.42 Betriebspraktikum

B159 Betriebspraktikum

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B159 |
| Bezeichnung | Betriebspraktikum |
| Lehrveranstaltung(en) | B159a Betriebspraktikum |
| Verantwortliche(r) | jeweiliger Dozent |
| Zuordnung zum Curriculum | Betriebswirtschaftslehre (Bachelor) Computer Games Technology (Bachelor) Data Science & Artificial Intelligence (Bachelor) E-Commerce (Bachelor) IT-Ingenieurwesen (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Smart Technology (Bachelor) Technische Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) Wirtschaftsingenieurwesen (Bachelor) |
| Verwendbarkeit | Studierende erweitern ihre sozialen Kompetenzen und ihre Kontakte zu Unternehmen. Beides können sie nach ihrem Studiumsabschluss gewinnbringend für eine Bewerbung oder das Einleben bei ihrem späteren Arbeitgeber verwenden. |
| Semesterwochenstunden | 0 |
| ECTS | 17.0 |
| Voraussetzungen | Keine |
| Dauer | 1 |

Lernziele

Die Studierenden sammeln Erfahrungen für die spätere berufliche Tätigkeit in einem Unternehmen.

Dies dient der Stärkung der beruflichen und sozialen Kompetenzen: Selbständigkeit, eigenverantwortliches Handeln, Kommunikation, Teamfähigkeit und Zeitmanagement.

I.1.42.1 Betriebspraktikum

| | |
|---------------------------------|-------------------------------|
| Lehrveranstaltung | Betriebspraktikum |
| Dozent(en) | jeweiliger Dozent |
| Hörtermin | 7 |
| Häufigkeit | jedes Semester |
| Lehrform | Betriebliches Praktikum |
| Semesterwochenstunden | 0 |
| ECTS | 17.0 |
| Prüfungsform | Praktikumsbericht / Protokoll |
| Sprache | deutsch |
| Lehr- und Medienform(en) | |

Lernziele

Das Betriebspraktikum ist ein zentraler Baustein für die berufliche Profilbildung der Studierenden. Es ermöglicht im Rahmen des Studiums einen direkten Kontakt zu Unternehmen, die von den Studierenden eigenständig angesprochen werden. Der Kontakt zum Unternehmen soll helfen, die im bisherigen Verlauf des Studiums angeeignete Fach- und Methodenkompetenz auf ausgewählte Abläufe und Problemstellungen des betrieblichen Alltags zu übertragen. Hierbei werden auch soziale Kompetenzen erprobt und gefestigt.

Inhalt

Das Betriebspraktikum soll vertieften Einblick in Prozesse und Aufbau eines Betriebes geben. Der oder die Studierende sucht sich das Betriebspraktikum mit Hilfe der Praktikadatenbank der Fachhochschule Wedel oder anderen Informationsquellen (z.B. Aushänge, Internetseiten des Wedeler Hochschulbundes). Bei Problemen bietet die Hochschule Hilfestellung. Die Tätigkeit kann im Rahmen des Tagesgeschäftes oder in einer Projektarbeit durchgeführt werden. Es wird aus Sicht der Hochschule angestrebt, dass das Betriebspraktikum als Vorlaufphase für eine sich unmittelbar anschließende Bachelorarbeit beim gleichen Unternehmen genutzt wird. Das Betriebspraktikum soll daher inhaltlich eine Brücke zur nachfolgenden Bachelorarbeit sein. Einsatzfelder sind in Absprache mit dem Unternehmen und dem oder der Dozent/in so zu wählen, das sie auch gut geeignet sind, eine Fragestellung für eine mögliche nachfolgende Bachelorarbeit zu entwickeln.

Literatur

themenabhängig

I.1.43 Bachelor-Thesis

B150 Bachelor-Thesis

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B150 |
| Bezeichnung | Bachelor-Thesis |
| Lehrveranstaltung(en) | B150a Bachelor-Thesis |
| Verantwortliche(r) | jeweiliger Dozent |
| Zuordnung zum Curriculum | Betriebswirtschaftslehre (Bachelor) Computer Games Technology (Bachelor) Data Science & Artificial Intelligence (Bachelor) E-Commerce (Bachelor) IT-Ingenieurwesen (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Smart Technology (Bachelor) Technische Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) Wirtschaftsingenieurwesen (Bachelor) |
| Verwendbarkeit | Keine. |
| Semesterwochenstunden | 0 |
| ECTS | 12.0 |
| Voraussetzungen | Voraussetzung ist das Wissen aus den Veranstaltungen der sechs vorangegangenen Semester, insbesondere der Veranstaltungen, die mit dem Themengebiet der Abschlussarbeit zusammenhängen. |
| Dauer | 1 |

Lernziele

An das Betriebspraktikum schließt sich die Bachelor-Arbeit an, die sehr praxisorientiert fast ausschließlich in Unternehmen angefertigt wird und deren Themenstellung sich in enger Kooperation zwischen FH Wedel und dem jeweiligen Unternehmen in der Regel aus dem betrieblichen Umfeld ergibt.

Die Studierenden sollen mit ihrer Arbeit den Nachweis erbringen, dass sie ihr erlerntes Wissen auf eine anwendungsbezogene Aufgabenstellung aus einem Fachgebiet selbstständig auf wissenschaftlicher Grundlage im Rahmen des festgelegten Themas anzuwenden. Wesentlich sind

strukturierte und argumentierte Inhalte sowie das Einhalten üblicher Formalia.

Dies dient der Vertiefung und des konkreten Einsatzes der fachliche Kompetenzen: Methodisches Arbeiten und praktisches Anwenden der im Studium erlernten Kenntnisse.

I.1.43.1 Bachelor-Thesis

| | |
|---------------------------------|--|
| Lehrveranstaltung | Bachelor-Thesis |
| Dozent(en) | jeweiliger Dozent |
| Hörtermin | 7 |
| Häufigkeit | jedes Semester |
| Lehrform | Thesis |
| Semesterwochenstunden | 0 |
| ECTS | 12.0 |
| Prüfungsform | Schriftl. Ausarbeitung (ggf. mit Präsentation) |
| Sprache | deutsch |
| Lehr- und Medienform(en) | |

Lernziele

Die Studierenden ...

- besitzen die Fähigkeit zur Durchführung einer praxisorientierten Arbeit.
- können eine Fragestellung selbständig erarbeiten.
- können die zu erarbeitende Problematik klar strukturieren.
- können die Vorgehensweise und Ergebnisse in einer Ausarbeitung übersichtlich darstellen.
- stärken ihre praktischen Fähigkeiten im Projektmanagement-Bereich und zur Selbstorganisation.

Inhalt

Die Bachelor-Thesis soll im Regelfall in Kooperation mit einem Unternehmen erarbeitet werden. Themen aus den Arbeitsgruppen und Laboren der Hochschule sind ebenfalls möglich. Die Arbeit ist als abschließende, vom Studierenden eigenständig aber hochschul- und unternehmensseitig betreutes Projekt zu verstehen. Im Sinne der Zielsetzung der Bachelor-Ausbildung, der Erlangung des ersten berufsqualifizierenden Abschlusses, ist die Arbeit thematisch an einer Problemstellung eines kooperierenden Unternehmens orientiert oder sie besteht aus einer praxisrelevanten hochschulinternen Aufgabe.

Literatur

themenabhängig

I.1.44 Bachelor-Kolloquium

B160 Bachelor-Kolloquium

| | |
|---------------------------------|---|
| Studiengang | Bachelor-Studiengang Informatik |
| Kürzel | B160 |
| Bezeichnung | Bachelor-Kolloquium |
| Lehrveranstaltung(en) | B160a Kolloquium |
| Verantwortliche(r) | jeweiliger Dozent |
| Zuordnung zum Curriculum | Betriebswirtschaftslehre (Bachelor) Computer Games Technology (Bachelor) Data Science & Artificial Intelligence (Bachelor) E-Commerce (Bachelor) IT-Ingenieurwesen (Bachelor) IT-Management, -Consulting & -Auditing (Bachelor) Informatik (Bachelor) Medieninformatik (Bachelor) Smart Technology (Bachelor) Technische Informatik (Bachelor) Wirtschaftsinformatik (Bachelor) Wirtschaftsingenieurwesen (Bachelor) |
| Verwendbarkeit | Keine. |
| Semesterwochenstunden | 1 |
| ECTS | 1.0 |
| Voraussetzungen | Zulassungsvoraussetzung zum Kolloquium ist eine mit mindestens "ausreichend" bewertete Bachelor-Thesis. |
| Dauer | 1 |

Lernziele

Das Kolloquium ist eine fächerübergreifende mündliche Prüfung, ausgehend vom Themenkreis der Bachelor-Thesis, und ist die letzte Prüfungsleistung, welche das Studium abschließt.

In der mündlichen Abschlussprüfung halten die Studierenden einen Fachvortrag über das von ihnen bearbeitete Thema und verteidigen ihre Bachelor-Thesis in einer anschließenden Diskussion. Dies stärkt die Fähigkeit, ein intensiv bearbeitetes Themengebiet, zusammenfassend darzustellen und professionell zu vertreten.

I.1.44.1 Kolloquium

| | |
|---------------------------------|-------------------|
| Lehrveranstaltung | Kolloquium |
| Dozent(en) | jeweiliger Dozent |
| Hörtermin | 7 |
| Häufigkeit | jedes Semester |
| Lehrform | Kolloquium |
| Semesterwochenstunden | 1 |
| ECTS | 1.0 |
| Prüfungsform | Kolloquium |
| Sprache | deutsch |
| Lehr- und Medienform(en) | |

Lernziele

Die Studierenden ...

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

Inhalt

- nach Thema der Bachelor-Arbeit unterschiedlich
- Fachvortrag über das Ergebnis der Bachelor-Arbeit
- Diskussion der Qualität der gewählten Lösung
- Fragen und Diskussion zum Thema der Bachelor-Arbeit und verwandten Gebieten

Literatur

themenabhängig

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