

| | | | | | |
|--|---|------------|------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Objektorientierte Softwareentwicklung | | Liste - | Modulnummer 1110 | Sem. 1 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 10 cp, 300 h 8 SWS | Lehrformen • Vorlesung (4 SWS) • Praktikum (2 SWS) • Übung (2 SWS) | | Häufigkeit jedes Jahr | Bewertung • Klausur oder mündl. Prüfung (70% als PL) • Praktische Tätigkeit als sep. SL (30% als SL) | |
| Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 30 h Teilnahme (Tutorium) • 60 h Vor- und Nachbereitung (Vorlesung) • 120 h Bearbeitung (Praktikumsaufgaben) | | | Dozenten Heimrich, Panitz | | Verantwortlich Panitz |
| Inhaltliche Voraussetzungen | | | | | |
| Lernziele • Objektorientierte Lösungen für einfache Problemstellungen finden, implementieren, dokumentieren und testen. • Algorithmisches Denken. • Praktischer Umgang mit Programmierwerkzeugen und Umgebungen. • Benutzung von Standardkomponenten und Bibliotheken in eigenen Problemlösungen | | | | | |
| Inhalte • Einführung (Softwareentwicklungszyklus, Algorithmus, Programm, Geschichte der Programmiersprachen) • Klassen als Abstraktionsebene mit Feldern, Methoden und Konstruktoren. • Vererbung • Schnittstellen • Kontrollfluss (Sequenz, Verzweigung, Schleifenbildung, Sprünge) • Operatoren und Ausdrücke (Vorrang, Assoziativität), konkrete Beispiele • Unterprogrammtechniken (Funktionen, Aufruf und Parameterübergabe, Laufzeitstack, Rekursion) • Ausnahmesituation • Generische Programmiertechniken | | | | | |
| Literatur Panitz, S. E.: Java will nur spielen, Vieweg Barnes, D. J. Kölling, M.: Java lernen mit BlueJ, Pearson Studium Heinisch, C.; Müller, F.; Goll, F.: Java als erste Programmiersprache, Teubner Krüger, G.: Handbuch der Java-Programmierung, Addison-Wesley Ullenboom, C.: Java ist auch eine Insel, Galileo Computing Oesterreich, B.: Analyse und Design mit UML, Oldenbourg | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter als PDF-Files | | | | | |

| | | | | | |
|---|--|------------|--------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Numerische Verfahren und Analysisgrundlagen | | Liste - | Modulnummer 1210 | Sem. 1 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Seminar (2 SWS) | | Häufigkeit jedes Jahr | Bewertung • Klausur oder mündl. Prüfung (80% als PL) • Praktische Tätigkeit als sep. SL (20% als SL) | |
| Aufwand • 60 h Bearbeitung (Übungsaufgaben) • 30 h Vor- und Nachbereitung (Vorlesung und Übungen) • 60 h Anwesenheit (Vorlesung und Übungen) | | | Dozenten Geib, Reith | | Verantwortlich Geib |
| Inhaltliche Voraussetzungen | | | | | |
| Lernziele Die Modellbildung und deren Analyse fordert von den Informatikern: <ul style="list-style-type: none"> • Verhalten und Eigenschaften von Funktionen untersuchen zu können • Fähigkeit Differential- und Integralrechnungen durchzuführen • gezielt verschiedene numerische Verfahren auszuwählen und einzusetzen • die Fähigkeit, verschiedene Verfahren nach gewissen Kriterien zu vergleichen • entstehende Fehler abzuschätzen | | | | | |
| Inhalte <ul style="list-style-type: none"> • Funktionen • Differentialrechnung • Funktionen mehrerer Variablen • Nullstellenbestimmung • Fehler • Interpolation • Approximation • Potenzreihenentwicklungen und Taylor-Polynome • Integralrechnung und Numerische Integration • zeitliche Integration | | | | | |
| Literatur L.Papula: Mathematik für Ingenieure, Vieweg, Wiesbaden, Band 1,2 Bronstein, Semendjajev: Taschenbuch der Mathematik V.P. Minorski: Aufgabensammlung der höheren Mathematik, Fachbuchverlag Leipzig, 2001 M. Knorrenschild: Numerische Mathematik, Fachbuchverlag Leipzig, 2003. J. Douglas Faires, Richard L. Burden: Numerische Methoden: Näherungsverfahren und ihre praktische Anwendung, Spektrum Akademischer Verlag GmbH, Heidelberg-Berlin-Oxford, 1993. | | | | | |
| Medienform / Unterrichtssprache <ul style="list-style-type: none"> • Skript • Folien, Übungsblätter und Lösungsblätter auf der Internetseite • Aufgabensammlung mit Lösungen zur Klausurvorbereitung | | | | | |

| | | | | | |
|--|--|------------|-------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Diskrete Strukturen | | Liste - | Modulnummer 1220 | Sem. 1 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Seminar (2 SWS) | | Häufigkeit jedes Jahr | Bewertung • Klausur oder mündl. Prüfung (80% als PL) • Praktische Tätigkeit als sep. SL (20% als SL) | |
| Aufwand • 60 h Bearbeitung (Übungsaufgaben) • 30 h Vor- und Nachbereitung (Vorlesung und Übungen) • 60 h Anwesenheit (Vorlesung und Übungen) | | | Dozenten Reith, Schwanecke | | Verantwortlich Reith |
| Inhaltliche Voraussetzungen | | | | | |
| <p>Lernziele</p> <p>Die Fähigkeit elementare mathematische Probleme zu lösen und einfache Erkenntnisse der Mathematik in der Informatik anzuwenden gehören zum Kern der Arbeit als Informatiker. Nach Beendigung dieses Moduls können die Studierenden:</p> <ul style="list-style-type: none"> • sicher mit den Grundbegriffen der mathematischen Logik umgehen und diese anwenden • beherrschen der wichtigsten Beweisverfahren und können diese auf einfach Problemstellungen selbstständig anwenden • können das Induktionsprinzip auf Objekte der Informatik (Graphen, Algorithmen, etc) anwenden • verstehen den Mengenbegriff und die Operationen auf Mengen • können Sachverhalte in geeigneten logischen Systemen formalisieren und mit diesen Formalisierungen in der Praxis umgehen • haben Kenntnisse grundlegender algebraischer Strukturen und ihrer Anwendungen in der Informatik erworben • verstehen die Grundprinzipien von asymmetrischen Kryptosystemen (RSA) <p>Die erworbenen Fähigkeiten tragen in besonderem Maße zur Ausprägung von formalen und mathematischen Kompetenzen bei, erweitern die Methodenkompetenzen und die Analyse-, Design- und Realisierungskompetenzen.</p> | | | | | |
| <p>Inhalte</p> <p>Logik</p> <ul style="list-style-type: none"> • Aussagen, Logische Verknüpfungen, Rechnen mit logischen Verknüpfungen • Aussageformen, Aussagen mit Quantoren • Beweise <p>Mengen</p> <ul style="list-style-type: none"> • Mengenoperationen, Potenzmenge, Kartesisches Produkt • Mächtigkeit von Mengen • Abzählbarkeit / Überabzählbarkeit <p>Relationen</p> <ul style="list-style-type: none"> • Funktionen • Ordnungen • Attribute (reflexiv, symmetrisch, transitiv, linear, surjektiv, injektiv, usw.) • Äquivalenzrelationen • Anwendung: kryptographische Hashfunktionen <p>Graphen</p> <ul style="list-style-type: none"> • gerichtet und ungerichtete Graphen, Adjazenzmatrix • Wege, Kreise, Zusammenhang <p>Induktion</p> <ul style="list-style-type: none"> • Prinzip der vollständige Induktion • Induktive Definitionen und strukturelle Induktion <p>Elementare Zahlentheorie und algebraische Strukturen</p> <ul style="list-style-type: none"> • Teilbarkeit, Kongruenzen • Gruppen, Ringe, Körper, Vektorräume • Anwendung: das asymmetrische Kryptosystem RSA | | | | | |
| <p>Literatur</p> <p>Haggarty, Diskrete Mathematik für Informatiker, Pearson Studium, 2004</p> <p>Meinel, Mundhenk, Mathematische Grundlagen der Informatik: Mathematisches Denken und Beweisen, Vieweg+Teubner, 2008</p> <p>Teschl, Teschl, Mathematik für Informatiker 1: Diskrete Mathematik und Lineare Algebra, Springer, 2008</p> | | | | | |

Medienform / Unterrichtssprache

- Veranstaltungsspezifische Web-Seite
- Skript/Folien und Übungsblätter als PDF

| | | | | | |
|--|--|------------|--|--|--|
| Modulbezeichnung / Prüfungsfach Technische Grundlagen der Informatik | | Liste - | Modulnummer 1230 | Sem. 1 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit jedes Jahr | Bewertung • Klausur oder mündl. Prüfung (70% als PL) • Praktische Tätigkeit als sep. SL (30% als SL) | |
| Aufwand • 60 h Bearbeitung (Praktikumsaufgaben) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Anwesenheit (Vorlesung und Praktikum) | | | Dozenten Gergeleit, Kaiser, Kröger, Werntges | | Verantwortlich Gergeleit |
| Inhaltliche Voraussetzungen | | | | | |
| Lernziele Nach Abschluss des Moduls haben die Studierenden <ul style="list-style-type: none"> • die grundlegenden Begriffen und Konzepten der technischen Informatik kennen gelernt, • verstehen die Prinzipien der Computerarchitektur und • verfügen über grundlegende Kenntnisse über den Aufbau und die Funktionsweise von Digitalrechnern und ihrer Teile. Die erworbenen Fähigkeiten tragen damit besonders zum Erwerb von fachspezifischen Analyse- und Designkompetenzen und zum Aufbau von spezifischen technologischen und Methodenkompetenzen bei. | | | | | |
| Inhalte <ul style="list-style-type: none"> • Schaltnetze (Schaltfunktionen, Schaltalgebra und Boolesche Gesetze, Normalformen) • Repräsentierung von Information in Rechensystemen (Bitfolgen, Zahlensysteme, Zahlendarstellungen, Arithmetik, Zeichenketten) • Computerarithmetik (Addier- und Multiplizier- und Dividierwerke) • Basisschaltungen (Multiplexer und Demultiplexer, Adressdecodierer) • Schaltwerke (verschiedene Flipflop-Typen, asynchrone und synchrone Schaltwerke, Dualzähler, Schieberegister, endliche Automaten, Speicher) • Architektur von Rechensystemen (Einführung und Überblick, von-Neumann-Architektur, Prozessorarchitektur, Systemarchitektur, Gerätekunde) • Grundlagen der Codierung (Einführung, Blockcodes, Codes variierender Länge, komprimierende Codes, fehlererkennende und -korrigierende Codes) | | | | | |
| Literatur Grundlagen der Technischen Informatik, Dirk W. Hoffmann Technische Informatik: Eine einführende Darstellung, B. Becker, P. Molitor | | | | | |
| Medienform / Unterrichtssprache <ul style="list-style-type: none"> • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter als PDF | | | | | |

| | | | | | |
|---|---------------------------------|--------------------------|--|--------------------------|--|
| Modulbezeichnung / Prüfungsfach Betriebswirtschaftslehre | | Liste - | Modulnummer 1310 | Sem. 1 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Seminar (4 SWS) | Häufigkeit jedes Jahr | Bewertung • Klausur oder mündl. Prüfung (100% als PL) | | |
| Aufwand • 30 h Bearbeitung (Übungsaufgaben) • 60 h Vor- und Nachbereitung (incl. Literaturstudium) • 60 h Anwesenheit (Seminaristischer Unterricht) | | | Dozenten Jarass | Verantwortlich Jarass | |
| Inhaltliche Voraussetzungen | | | | | |
| Lernziele Die Kernausbildung in der Informatik und die breite Grundlagenausbildung in Mathematik, Technik, Wirtschaft und Recht bereitet die Absolventen für ihre Aufgaben in der Praxis vor und bildet die Grundlage für die kontinuierliche Weiterbildung im Berufsleben. Das Informatikstudium wird durch die Ausbildung in betriebswirtschaftlichen und juristischen Grundlagen ergänzt. Die Studierenden sollen befähigt werden • die Grundlagen wirtschaftlichen Handelns zu erkennen, • betriebliche Zusammenhänge zu analysieren und zu beeinflussen. • wirtschaftliche Grundbedingungen und Zusammenhänge der betrieblichen Aktivitäten zu beurteilen, • Kosten von Projekten abzuschätzen und zu kontrollieren, • Investitions- und Projektkalkulationen durchzuführen. | | | | | |
| Inhalte Die Studierenden lernen ausgewählte Grundlagen der BWL und können diese auf den Einsatz von IT-Systemen übertragen. • Grundbegriffe betriebswirtschaftlicher Analyse: Grundfunktionen, Produktionsfaktoren, Kennzahlen • Vertrieb • Investition und Finanzierung • Computerunterstützung im Unternehmen: Einsatz und Bedeutung von IT-Lösungen • Rechnungswesen: Kostenarten-, Kostenstellen-, Kostenträgerrechnung • Unternehmensführung: Organisation, Controlling | | | | | |
| Literatur Olfert, Klaus und Rahn, Hans-Jürgen: Einführung in die Betriebswirtschaftslehre. 10. Auflage, Kiehl-Verlag, 2010. Händler, Jürgen (Hrsg.): Betriebswirtschaftslehre für Ingenieure. Lehr- und Praxisbuch. Hanser Technikbücher. 4., aktualisierte Auflage 2010. | | | | | |
| Medienform / Unterrichtssprache • Lehrbuch, Beamer, Tafelanschrieb. Weitere Informationen unter www.JARASS.com , Lehre | | | | | |

| | | | | | |
|--|---|------------|------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Programmiermethoden und -techniken | | Liste - | Modulnummer 2110 | Sem. 2 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 10 cp, 300 h 8 SWS | Lehrformen • Vorlesung (4 SWS) • Praktikum (2 SWS) • Übung (2 SWS) | | Häufigkeit jedes Jahr | Bewertung • Klausur oder mündl. Prüfung (70% als PL) • Praktische Tätigkeit als sep. SL (30% als SL) | |
| Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 30 h Teilnahme (Tutorium) • 60 h Vor- und Nachbereitung (Vorlesung) • 120 h Bearbeitung (Praktikumsaufgaben) | | | Dozenten Heimrich, Panitz | | Verantwortlich Panitz |
| Inhaltliche Voraussetzungen • Modul "Objektorientierte Software-Entwicklung" | | | | | |
| Lernziele Die Studierenden kennen ein umfassendes Instrumentarium an Techniken und Lösungsmustern zur Softwareentwicklung in unterschiedlichen Programmiersprachen. Die Fähigkeit zum Entwickeln von eigenen Bibliotheken und komplexen Anwendungen in einzelnen Komponenten wurde erworben. Standardarchitekturmuster werden beherrscht. | | | | | |
| Inhalte • Baum-, Ereignis- oder Strom-basierte Verarbeitung Hierarchischer Strukturen • Speicherbelegung und Freigabe, explizit und implizit • Nebenläufigkeit • Kommunikation • Funktionen als Typen für Argumente • Modell-View-Control Architekturen | | | | | |
| Literatur Ullенboom, C.: Java ist auch eine Insel, Galileo Computing Scholz, M., Niedermeier, S.: Java und XML Kernighan, B., Ritchie, D.: Programmieren in C. ANSI C Erlenkötter, H.: C Programmieren von Anfang an | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter als PDF-Files. | | | | | |

| | | | | | |
|---|--|------------|---------------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Algorithmen und Datenstrukturen | | Liste - | Modulnummer 2120 | Sem. 2 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit jedes Jahr | Bewertung • Klausur oder mündl. Prüfung (70% als PL) • Praktische Tätigkeit als sep. SL (30% als SL) | |
| Aufwand • 60 h Bearbeitung (Praktikumsaufgaben) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Anwesenheit (Vorlesung und Praktikum) | | | Dozenten Heimrich, Reith, Werntges | | Verantwortlich Reith |
| Inhaltliche Voraussetzungen | | | | | |
| Lernziele Entwurf, Implementierung und Auswahl von Algorithmen und Datenstrukturen für gegebene Problemstellungen sind typische Aufgaben eines Informatikers. Nach Beendigung dieses Moduls können die Studierenden <ul style="list-style-type: none"> • selbstständig Algorithmen entwerfen, bewerten (Laufzeit) und implementieren • dynamische Datenstrukturen (objektorientiert) implementieren • für Problemstellungen passende Algorithmen und Datenstrukturen auswählen und bestehende Bibliotheken nutzen • graphentheoretische Konzepte für praktische Problemstellungen anwenden Die erworbenen Fähigkeiten tragen in besonderem Maße zur Ausprägung von algorithmischen Kompetenzen bei, erweitern die Methodenkompetenzen und die Analyse-, Design- und Realisierungskompetenzen. | | | | | |
| Inhalte <ul style="list-style-type: none"> • Eigenschaften von Algorithmen, Probleme versus Algorithmen • Suchen, einfache Sortierverfahren, effiziente Sortierverfahren • Laufzeit und Komplexität, O-Notation, Analyse von Algorithmen, Lösen von Rekurrenzen • Algorithmenentwurf und Algorithmenmuster • Abstrakte Datentypen und deren Implementierung (Listen, Mengen) • Einfache dynamische Datenstrukturen (verkettete Listen, Keller, Warteschlangen) • Bäume, Durchlaufen, Binärbäume, Suchbäume, Ausgeglichene Bäume • Hashing, Hash-Funktionen, Kollisionsbehandlung • Graphen | | | | | |
| Literatur Saake, Sattler: Algorithmen und Datenstrukturen in Java, dpunkt.verlag, 2006 Cormen, Leiserson, Rivest, Stein, Introduction to Algorithms, MIT Press, 2001 Sedgewick: Algorithmen in C, Addison-Wesley, 1993 Ottmann, Widmayer: Algorithmen und Datenstrukturen, Spektrum, 2002 | | | | | |
| Medienform / Unterrichtssprache <ul style="list-style-type: none"> • Veranstaltungsspezifische Web-Seite • Skript, Folien und Übungsblätter | | | | | |

| | | | | | |
|---|--|------------|--------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Datenbanken | | Liste - | Modulnummer 2130 | Sem. 2 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit jedes Jahr | Bewertung • Klausur oder mündl. Prüfung (70% als PL) • Praktische Tätigkeit als sep. SL (30% als SL) | |
| Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben) | | | Dozenten Muth | Verantwortlich Muth | |
| Inhaltliche Voraussetzungen | | | | | |
| Lernziele Datenbanksysteme zum konsistenten Verwalten von Daten für Anwendungen im Ein- und Mehrbenutzerbetrieb sind aus der Praxis nicht mehr wegzudenken und werden von jedem Informatiker betrieben und genutzt. • Modellieren, Erstellen von Entity-Relationship-Modellen und Umsetzung in relationale Datenmodelle • Datenunabhängigkeit, Relationenmodell und -algebra, Transaktionskonzept verstehen und einsetzen können • Kenntnisse in SQL zur Datendefinition, Datenmanipulation und für Anfragen nutzen, relationalen Datenbanken von Programmiersprachen aus nutzen | | | | | |
| Inhalte • Daten, Datenmodellierung, Datenunabhängigkeit, logische Datenmodelle • Architektur von Datenbanksystemen, Schemaarchitektur, Systemarchitektur • Entity-Relationship (ER) Modell (Entity-Typen, Attribute, Beziehungen), Kardinalitäten, Erweiterungen • Relationales Modell, Schlüssel (Primärschlüssel, Fremdschlüssel), referentielle Integrität • Relationenalgebra (Projektion, Selektion, Verbunde, Umbenennung), Vollständigkeit, Kalkül • SQL (Datendefinition, Datenmanipulation, Anfragen), Sichten, Domänen, SQL-Erweiterungen • Datenbankdesign, Umsetzung ER-Modell in relationales Modell • Funktionale Abhängigkeiten, Normalformen, Normalisierung • Einbettung von Programmiersprachen in SQL (Imperative Sprachen, Trigger), Einbettung von SQL in Programmiersprachen (Cursor-Konzept) • Transaktionskonzepte (Atomar, Konsistent, Isolation, Dauerhaftigkeit), Serialisierung, Sperrkonzepte, Logging, Recovery • Sicherheitsmodelle, Rechtevergabe, Rechtevergabe bei Sichten • Anfrageoptimierung (Index, Analysetools), Betrieb, Wartung | | | | | |
| Literatur Saake, Heuer: Datenbanken - Konzepte und Sprachen, mitp Kemper, Eickler: Datenbanksysteme - Eine Einführung, Oldenbourg Verlag Faeskorn-Woyke, Bertelsmeier, Riemer, Bauer: Datenbanksysteme -Theorie und Praxis mit SQL2003, Oracle und MySQL, Pearson Studium | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter | | | | | |

| | | | | | |
|---|--|------------|-----------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Rechnernetze und Telekommunikation | | Liste - | Modulnummer 2140 | Sem. 2 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit jedes Jahr | Bewertung • Klausur oder mündl. Prüfung (70% als PL) • Praktische Tätigkeit als sep. SL (30% als SL) | |
| Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben) | | | Dozenten Geib, Gergeleit | | Verantwortlich Gergeleit |
| Inhaltliche Voraussetzungen • Modul Technische Grundlagen der Informatik: Grundlagen der Codierung • Objektorientierte Softwareentwicklung: Imperatives Programmieren, Umgang mit Programmierungswerkzeugen | | | | | |
| Lernziele Die Integration von klassischen Telekommunikation-, Daten- und Mobilfunknetzen bildet die Grundlage der globalen Vernetzung und damit die Basis praktisch aller zukünftigen Anwendungen. Die grundlegende Kenntnis des Aufbaus und der Konzepte der Vernetzung ist daher notwendig zur Nutzung, Erstellung und Bewertung von informationstechnischen Systemen in allen Bereichen. Nach erfolgreichem Abschluss dieser Lehrveranstaltung haben die Studierenden erworben: • Verständnis von Prinzipien und Teilaufgaben der Schichten und ihrer Protokolle • Kompetenz die Eignung von Netzwerktechnologien und Protokollen bei gegebenen Anwendungen / Anforderungen beurteilen zu können • insbesondere Kenntnisse über die Internet-Protokolle und ihrer Eigenschaften • Wissen über die Grundlagen und Methoden der Netzwerksicherheit und des Netzwerkdesigns • Fähigkeiten Dienste auf Basis des Socket-APIs konzipieren und realisieren zu können Die erworbenen Fähigkeiten tragen damit in besonderem Maße zur Ausprägung von Netzwerk-spezifischen Analyse-, Design- und Realisierungskompetenzen und zur Erweiterung von spezifischen technologischen Kompetenzen bei. | | | | | |
| Inhalte • Schichtenmodelle (insb. das ISO / OSI 7-Schichten-Modell) • Anwendungsschicht (Aufgaben, Typische Protokolle für verschiedene Anwendungen wie z.B. Mail, Web, Dienste wie z.B. DNS) • Transportschicht (Aufgaben, Dienste, Protokolle TCP / UDP, Adressierung, Methoden zur Zuverlässigkeit, zur Stau- und Flußkontrolle bei TCP) • Vermittlungsschicht (Aufgaben, Dienste der Vermittlungsschicht, Adressierung, IPv4/IPv6, Subnetting, NAT, Routing, Algorithmen zur Wegbestimmung, Routing-Protokolle) • Sicherungsschicht (Aufgaben der Sicherungsschicht; Fehlererkennung und -korrektur, Flußkontrolle) • Mehrfachzugriffskontrolle (LAN-Adressierung und ARP, Beispiele wie Ethernet, IEEE 802.11 WLANs, Komponenten (Hubs, Switches, Bridges), STP) • Netzwerkplanung und Netzwerkmanagement • Netzwerksicherheit (Schutzziele und Bedrohungen, Schutzmaßnahmen, Krypto-Algorithmen, Protokolle, Sicherheitsarchitekturen) | | | | | |
| Literatur Computernetzwerke, Andrew S. Tanenbaum Computernetze - Ein Top-Down-Ansatz, James F. Kurose, Keith W. Ross | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter als PDF | | | | | |

| | | | | | |
|--|--|------------|--------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Lineare Algebra | | Liste - | Modulnummer 2210 | Sem. 2 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Seminar (2 SWS) | | Häufigkeit jedes Jahr | Bewertung • Klausur oder mündl. Prüfung (80% als PL) • Praktische Tätigkeit als sep. SL (20% als SL) | |
| Aufwand • 60 h Anwesenheit (Vorlesung und Übungen) • 30 h Vor- und Nachbereitung (Vorlesung und Übungen) • 60 h Bearbeitung (Übungsaufgaben) | | | Dozenten Geib, Reith | | Verantwortlich Reith |
| Inhaltliche Voraussetzungen | | | | | |
| Lernziele Die Modellbildung und deren Analyse fordert von den Informatikern den sicheren Umgang mit Vektoren und linearen Abbildungen • Beherrschung der Matrizenrechnung • Lösen von linearen Gleichungssystemen • Umgang mit komplexen Größen • Berechnung der Schrittpunkte, Schnittwinkel und Abstände von Geraden und Ebenen | | | | | |
| Inhalte • Vektoren in zweidimensionalen Raum • Komplexe Zahlen • Vektorrechnung im dreidimensionalen Raum • Der allgemeine n-dimensionale Vektorraum • Lineare Abbildungen und Matrizen • Lineare Gleichungssysteme • Numerische Lösung linearer Gleichungssysteme • Die Inverse einer quadratischen Matrix • Die Determinante einer Matrix • Das Rechnen mit Matrizen • Lineare Kombinationen, lineare Unabhängigkeit, Basis und Dimension • Eigenwerte und Eigenvektoren. • Parameterdarstellung von Geraden, Ebenen, Normalform • Übergang von einem Koordinatensystem zu einem anderen, Basiswechsel | | | | | |
| Literatur Papula: Mathematik für Ingenieure und Naturwissenschaftler, Band 1 und 2, Vieweg, 2006 Bronstein, Semendjajev: Taschenbuch der Mathematik, Deutsch (Harri), 2008 Gramlich: Lineare Algebra: Eine Einführung, Hanser, 2009 Gramlich: Lineare Algebra: Fachbuchverlag Leipzig, 2003 Farin, Hansford: Lineare Algebra - ein geometrischer Zugang, Springer, 2003 | | | | | |
| Medienform / Unterrichtssprache • Skript • Übungsblätter und Lösungsblätter auf der Internetseite | | | | | |

| | | | | | |
|---|--|------------|------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Webbasierte Anwendungen | | Liste - | Modulnummer 3110 | Sem. 3 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit jedes Jahr | Bewertung • Klausur oder mündl. Prüfung (70% als PL) • Praktische Tätigkeit als sep. SL (30% als SL) | |
| Aufwand • 60 h Bearbeitung (Praktikumsaufgaben) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Anwesenheit (Vorlesung und Praktikum) | | | Dozenten Martin, Werntges | | Verantwortlich Martin |
| Inhaltliche Voraussetzungen | | | | | |
| Lernziele Webbasierte Systeme sind eine der verbreitetsten Ausprägungen verteilter Informationssysteme und sind heute weder aus dem privaten noch aus dem beruflichen Bereich wegzudenken. Solche Systeme professionell planen, realisieren und beurteilen zu können, stellt daher eine wichtige Qualifikation dar. • Kennen typischer Anwendungsgebiete webbasierter Anwendungen • Fähigkeit zum problemadäquaten Entwurf und zur Realisierung webbasierter Anwendungen, einschl. Beurteilung und Auswahl geeigneter Basistechnologien • Sensibilisierung für das Gefahrenpotenzial, Kenntnis grundlegender Abwehrmaßnahmen | | | | | |
| Inhalte • Einführung, Klassifikation von Web-Anwendungen, Architekturen • Grundlagen (HTTP, Session-Management, Standardisierung, W3C) • Webseiten gestalten (HTML, Cascading Stylesheets, Web-Usability) • Serverseitige Technologien: Skriptsprachen, Template-Technik, Servlets, JSP • Clientseitige Technologien: JavaScript, Ajax, DOM • Mehrschichtenarchitekturen, Frameworks zu deren Umsetzung • Applikationsserver (Aufgaben, Dienste) • Sicherheitsaspekte: TLS und Zertifikate, Bedrohungen wie SQL Injection, XSS, session hijacking; Gegenmaßnahmen | | | | | |
| Literatur Günter Pomaska: "Grundkurs Web-Programmierung", Vieweg 2005 Günther Bauer: "Architekturen für Web-Anwendungen" Vieweg+Teubner 2009 Martin Marinschek: "JSF @ Work", dpunkt 2007 Donald Brown: "Struts 2 im Einsatz", Hanser 2008 Sam Ruby: "Agile Web Development with Rails", 4. ed., Pragmatic Bookshelf 2011 Mario Heiderich, Christian Matthies, Johannes Bahse, Fukami: "Sichere Webanwendungen", 1. Auflage, Galileo Computing, 2009 | | | | | |
| Medienform / Unterrichtssprache • Web-Seite zur Veranstaltung • Folien, Übungsblätter | | | | | |

| | | | | | |
|---|--|------------|--------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Softwaretechnik | | Liste - | Modulnummer 3120 | Sem. 3 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit jedes Jahr | Bewertung • Klausur oder mündl. Prüfung (70% als PL) • Praktische Tätigkeit als sep. SL (30% als SL) | |
| Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben) | | | Dozenten Iglter | | Verantwortlich Iglter |
| Inhaltliche Voraussetzungen | | | | | |
| Lernziele Die Fähigkeit zur Auswahl, Bewertung und praktischen Anwendung von Konzepten und Methoden zur systematischen Entwicklung (großer) Softwaresystemen stellt eine zentrale Qualifikation für Informatiker dar. Dabei sind die Phasen Analyse / Design von grundlegender Bedeutung für das Gelingen eines Softwareprojekts: • Vorgehensmodelle, Softwareentwicklung im Team • Modellierung, Einsatz der Unified Modeling Language (UML) • Nutzung von Softwarewerkzeugen (CASE-Tools) | | | | | |
| Inhalte • Einführung; Entstehung und Entwicklung der Disziplin "Softwaretechnik" • Phasen der Softwareentwicklung • Vorgehensmodelle (Wasserfall, Spiralmodell; Vergleich) • Agile Prozesse (z.B. eXtreme Programming) als Beispiel für einen "leichtgewichtigen Prozess"; Eignung / Vor- und Nachteile agiler Prozesse • Rolle der Modellierung in der SW-Entwicklung • Rational Unified Process als Beispiel für einen "schwergewichtigeren", inkrementellen und iterativen Softwareentwicklungsprozess • Analysephase (Ziele, Dokumente (insb. Lasten-/Pflichtenheft), Modellierung: Domänen- und Analyseklassendiagramme, ggf. Ablaufmodellierung, Vorgehensweisen; Prototyping; Validierung der Analyseergebnisse) • Sprachelemente der UML: Use-Case-Diagramme, Klassendiagramme, Diagrammtypen zur Verhaltensmodellierung • Entwurfsphase (SW-Architekturbegriff, Kohäsion/Kopplung, Verantwortlichkeiten, Muster-Begriff; Architektur- / Verteilungs- / Entwurfsmuster; Entwurfsmodellierung; Dokumente der Entwurfsphase; spezielle OO-Fragestellungen) • Entwurfsmuster ("Gang of Four"; Verhaltens-, Struktur- und Erzeugungsmuster) • Wiederverwendung (Arten der Wiederverwendung; Softwarekomponenten) • Einsatz von CASE-Tools • Implementierungsphase | | | | | |
| Literatur Stephan Kleuker: "Grundkurs Software-Engineering mit UML", Vieweg+Teubner 2009 Oestereich: "Analyse und Design mit UML 2.1, Oldenbourg Verlag, 2006 Eric Freeman, Elisabeth Freeman, Kathy Sierra, Bert Bates: "Entwurfsmuster von Kopf bis Fuß", Vieweg +Teubner, GWV-Fachverlage Wiesbaden 2009 | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Folien, Übungsblätter | | | | | |

| | | | | | |
|---|--|------------|--|--|--|
| Modulbezeichnung / Prüfungsfach Computergrafik | | Liste - | Modulnummer 3130 | Sem. 3 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit jedes Jahr | Bewertung • Klausur oder mündl. Prüfung (70% als PL) • Praktische Tätigkeit als sep. SL (30% als SL) | |
| Aufwand • 60 h Bearbeitung (Praktikumsaufgaben) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Anwesenheit (Vorlesung und Praktikum) | | | Dozenten Dörner, Schulz, Schwanecke | | Verantwortlich Schulz |
| Inhaltliche Voraussetzungen • Objektorientierte Softwareentwicklung: Programmierkenntnisse in einer imperativen Programmiersprache • Lineare Algebra: Vektoren, Matrizen, Eigenwerte, Eigenvektoren, Determinante, Lineare Gleichungssysteme | | | | | |
| Lernziele • Verständnis des Renderingprozesses und Kenntnis des aktuellen Stands der Technik bezüglich der Renderingschritte • Sicherer Umgang mit für die Computergrafik relevanten Konzepten aus dem Bereich der analytischen Geometrie • Kenntnis verschiedener Objektrepräsentationen und zugehöriger Datenstrukturen • Praktische Erfahrungen mit einer Grafik-API (z.B. OpenGL, DirectX, ...) und einer Szenengraph-API (z.B. OpenInventor, Java3D, ...) Die erworbenen Fähigkeiten tragen damit in besonderem Maße zur Ausprägung von spezifischen Analyse-, Design- und Realisierungskompetenzen und zur Erweiterung von spezifischen technologischen Kompetenzen bei. | | | | | |
| Inhalte • Objektdarstellungen in 2D/3D (explizite, implizite, parametrische Darstellungen) • Netze (Polygon-Netze, Dreiecks-Netze, Datenstrukturen) • Freiform-Kurven und -Flächen (Bezier-, B-Spline-, Unterteilungs-Kurven und Flächen) • Geometrische Grundlagen (lineare, affine, projektive Transformationen in homogenen Koordinaten, Quaternionen) • Projektionen (Parallel-, Zentral-, Stereo-Projektion) • Darstellungen von Primitiven auf Rasterbildschirmen (Linien, Polygone, gefüllte Bereiche) • Grundlagen Farben und Texturen • Renderpipeline (Polygon-Clipping, -Rasterisierung, -Triangulation, Beleuchtung/Shading, Texturierung, Verdeckung (Maler, z-Buffer), Vertex-, Geometrie- und Fragment-Shader) • Spezielle Render-Techniken (Schattenberechnung, Environment-Mapping,...) • Lokale Beleuchtungsmodelle (Phong-Modell, Abschwächung, Spotlicht, Depth-Cueing, BRDFs) • Globale Beleuchtungsverfahren (Ray-Casting, Ray-Tracing) • Standard Grafik-APIs (OpenGL, DirectX, ...) • Szenengraph-APIs (OpenInventor, Java3D, ...) • Anwendungen der Computergrafik (AR, VR, ...) | | | | | |
| Literatur Foley, van Dam, Feiner, Hughes: Computer Graphics: Principles and Practise, Addison Wesley Watt: 3D-Computergrafik, Pearson Studium Woo, Neider, Davis: OpenGL, Programming Guide, The Official Guide to Learning OpenGL, Addison Wesley Angel, Interactive Computer Graphics, Addison Wesley | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript, Folien und Übungsblätter (als pdf-Dateien) | | | | | |

| | | | | | |
|---|---|------------|---------------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Betriebssysteme und Rechnerarchitektur | | Liste - | Modulnummer 3140 | Sem. 3 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 10 cp, 300 h 8 SWS | Lehrformen • Vorlesung (4 SWS) • Seminar (2 SWS) • Praktikum (2 SWS) | | Häufigkeit jedes Jahr | Bewertung • Klausur oder mündl. Prüfung (70% als PL) • Praktische Tätigkeit als sep. SL (30% als SL) | |
| Aufwand • 120 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Vorlesung) • 120 h Bearbeitung (Praktikumsaufgaben) | | | Dozenten Gergeleit, Kaiser, Kröger | | Verantwortlich Kröger |
| Inhaltliche Voraussetzungen • Modul Programmiermethodik und Programmieretechniken: Programmieren in C, Java • Technische Grundlagen der Informatik: Repräsentierung von Information, Architektur von Rechensystemen | | | | | |
| Lernziele Die Kenntnis zentraler Betriebssystemkonzepte und das Wissen um die darunterliegenden HW-Strukturen ist für ein solides Gesamtverständnis moderner IT-Systeme unerlässlich und eine wesentliche Voraussetzung für viele Berufsfelder eines Informatikers, insb. in der systemnahen Programmierung, der Systemadministration und der Computersicherheit • Die Studierenden kennen die grundlegenden Konzepte moderner Rechnersysteme und deren Betriebssysteme. • Sie können Probleme auf hardwarenahen Abstraktionsebenen analysieren und lösen und • Strategien zur Performanceverbesserung und zur verbesserten Systemsicherheit anwenden. • Sie beherrschen wesentliche Teile der UNIX-Programmierschnittstelle im praktischen Umgang. Die erworbenen Fähigkeiten tragen damit in besonderem Maße zur Ausprägung von System-spezifischen Analyse- und Designkompetenzen und darüber hinaus zum Erwerb von Hardware-spezifischen technologischen Kompetenzen sowie zu den Realisierungskompetenzen systemnaher Software bei. | | | | | |
| Inhalte • Einführung (Historische Entwicklung der Betriebssysteme und Rechnerarchitekturen, Schichtenmodell der Rechnerorganisation) Betriebssysteme: • Betriebssystemkonzepte (Architekturen, Virtualisierung, Einsatzbereiche) • Prozessverwaltung (Prozesskonzept, Threads, Scheduling) • Prozesssynchronisation (Concurrency, Race Conditions, Wechselseitiger Ausschluss, Synchronisationsmechanismen) • Prozesskommunikation (nachrichten- und speicherorientierte Kommunikationsmechanismen, Performance-Betrachtungen) • Deadlocks (Grundlagen, Verfahren zur Problembehandlung) • Speicherverwaltung (Swapping, Virtual Memory Management, Seitenersetzungsalgorithmen) • Ein-/Ausgabe (I/O-Software, Treiber, insb. Plattentreiber, Uhrtreiber) • Dateisysteme (Dateien, Verzeichnisse, Dateisystemtypen, Fehlertoleranz, Datensicherung, Performance-Betrachtungen) • Sicherheit (Schutzmechanismen, Authentifikation, Autorisierung, vertrauenswürdige Systeme, Klassifizierungen) Rechnerarchitektur: • Maschinencode-Ebene (Instruktionen, Adressierung, Sprünge, Unterprogramme, Stacks, Parameterübergabe, Systemaufrufe, Interrupts) • Assemblersprache mit Beispielen • Prozessorarchitektur mit Optimierungen (Vergleich RISC/CISC, Pipelining, Branch-Prediction, Out-of-Order-Execution, Leistungsbewertung) • Speicherarchitektur (virtueller Speicher, MMU-Organisation und TBL, Page-Tables, Caches, Speicherhierarchien) • Multiprozessoren (Kommunikationsmodelle, Verbindungsnetzwerke, Cache-Kohärenz) • I/O-Interfacing (Geräte-Klassen, I/O-Ports, I/O-Busse, Arbitrierung, DMA) • Sicherheit (Speicherschutz, Exploit-Techniken, Schutzmechanismen) | | | | | |
| Literatur Patterson, Hennessy: Rechnerorganisation und -entwurf, 3. Auflage, Spektrum Akademischer Verlag, 2005 Tanenbaum: Computerarchitektur. Strukturen - Konzepte - Grundlagen, Pearson Studium, 2005 Tanenbaum: Moderne Betriebssysteme, 3. aktualisierte Auflage, Pearson Studium, 2009 Stallings: Operating Systems - Internals and Design Principles, 6th Ed., Pearson, 2009 | | | | | |

Medienform / Unterrichtssprache

- Veranstaltungsspezifische Web-Seite
- Skript/Folien und Übungsblätter als PDF
- Ergänzende Online-Selbstlernmodule

| | | | | | |
|--|--|------------|----------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Verteilte Systeme | | Liste - | Modulnummer 3150 | Sem. 3 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit jedes Jahr | Bewertung • Klausur oder mündl. Prüfung (70% als PL) • Praktische Tätigkeit als sep. SL (30% als SL) | |
| Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben) | | | Dozenten Kröger, Schmid | | Verantwortlich Kröger |
| Inhaltliche Voraussetzungen • Modul Programmiermethodik und Programmieretechniken: Programmieren in Java, C | | | | | |
| Lernziele Die heutige Entwicklung moderner verteilter Anwendungen nutzt verschiedene Paradigmen verteilter Anwendungen und verwendet vorhandene standardisierte Dienste. • Grundlagen und Strukturen verteilter Systeme kennen und beurteilen können • Verteilte Dienstumgebungen kennen und beurteilen können • Paradigmen der Programmierung verteilter Anwendungen kennen und anwenden können • Neue Problemstellungen für verteilter Anwendungen analysieren und mit bekannten Paradigmen unter Nutzung einer Dienstumgebung lösen können | | | | | |
| Inhalte • Verteilte Systemarchitekturen (HW-Systemstrukturen, SW-Grundstrukturen [verteilte Programme, verteilte Kontrolle, Transparenzarten, Netzwerkbetriebssysteme, Verteilte Betriebssysteme], Middleware-Konzepte, Architekturen für Hochverfügbarkeit, Cloud Computing) • Nachrichtenorientierte Kommunikation (Kooperationsmodelle [Client/Server, Multi-Tier, Peer-to-Peer, Gruppenkommunikation], verteilte und parallele Anwendungen, Message-Oriented Middleware) • Dienstorientierung (Remote Procedure Calls [Grundprinzip, Binding, Parameterbehandlung, Semantik im Fehlerfall, Sicherheit, RPC-Protokolle], Beispiele, Dienstumgebungen (Grundlagen, Architektur, Beispiele) • Objektorientierte Middleware (Grundlagen, CORBA, CORBA Services, Interoperabilität, Beispiele) • Service-orientierte Architekturen (Prinzip, Web Services, SOAP, WSDL, REST-Architekturstil) • Spezielle Dienste (Namens/Trader-Dienste, globale Zeitdienste, verteilte Dateidienste, Transaktionssteuerungsdienste, Authentifizierungs- und Autorisierungsdienste) | | | | | |
| Literatur Tanenbaum, van Steen: "Verteilte Systeme - Grundlagen und Paradigmen", Pearson Studium, 2. Auflage, 2007 Coulouris, Dollimore, Kindberg, Blair: "Distributed Systems - Concepts and Design", Pearson Studium, 5. Auflage, 2012 | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter als PDF • Ergänzende Online-Selbstlernmodule (Wissenswerkstatt Rechensysteme) | | | | | |

| | | | | | |
|---|--|------------|--------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Automatentheorie und Formale Sprachen | | Liste - | Modulnummer 4110 | Sem. 4 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Seminar (2 SWS) | | Häufigkeit jedes Jahr | Bewertung • Klausur oder mündl. Prüfung (80% als PL) • Praktische Tätigkeit als sep. SL (20% als SL) | |
| Aufwand • 60 h Bearbeitung (Übungsaufgaben) • 30 h Vor- und Nachbereitung (Vorlesung und Übungen) • 60 h Anwesenheit (Vorlesung und Übungen) | | | Dozenten Barth, Geib, Reith | | Verantwortlich Reith |
| Inhaltliche Voraussetzungen | | | | | |
| Lernziele Logisches Denken, Beweistechniken und strukturiertes Vorgehen - unabhängig von konkreten Rechnern und aktuellen Trends - ist Grundlage solider konzeptioneller Arbeit. Nach Beendigung dieses Moduls können die Studierenden: <ul style="list-style-type: none"> • Verfahren zur Mustererkennung praktisch (z.B. für die Suche in Texten, Syntexanalyse und Kodierung) anwenden • haben Erkenntnisse über grundsätzliche und praktische Lösbarkeit eines Problems erworben und können diese auf neue Probleme übertragen • selbstständig Überlegungen über praktische Aufgabenstellungen auf die gefestigten theoretischen Grundlagen der Informatik aufbauen • Möglichkeiten und Grenzen von (zukünftigen) Technologien einschätzen Die erworbenen Fähigkeiten tragen in besonderem Maße zur Ausprägung von formalen und mathematischen Kompetenzen bei, erweitern die Methodenkompetenzen und die Analyse-, Design- und Realisierungskompetenzen. | | | | | |
| Inhalte <ul style="list-style-type: none"> • Grundlegende Begriffe, Semi-Thue-Systeme, L-Systeme, Chomsky-Grammatiken, Chomsky-Hierarchie, Wortproblem • Deterministische/Nicht-deterministische endliche Automaten, Äquivalenz und Minimierung, Reguläre Sprachen, Äquivalenz zu endlichen Automaten, Operationen und Abschlusseigenschaften, Pumping-Lemma • Kontextfreie Sprachen, Mehrdeutigkeit, Chomsky-Normalform, Pumping-Lemma, CYK-Algorithmus, Deterministische/Nicht-deterministische Kellerautomaten, Äquivalenz von Kellerautomaten und kontextfreien Grammatiken • Kontextsensitive- und Typ0-Sprachen, Turing-Maschinen • Turing-Berechenbarkeit, Gödelisierung, Berechenbarkeit und Entscheidbarkeit, Halteproblem • Nicht handhabbare Probleme, Komplexität, Problemklassen P und NP, NP-vollständige Probleme, Umgang mit NP-vollständigen Probleme in der Praxis | | | | | |
| Literatur Hopcroft, Ullman, Motwani, Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie, Pearson, 2002 Schöning, Theoretische Informatik - kurz gefasst, Spektrum, 2008 Michael Sipser, Introduction to The Theory of Computation, Thomson Press, 2005 | | | | | |
| Medienform / Unterrichtssprache <ul style="list-style-type: none"> • Veranstaltungsspezifische Web-Seite • Skript, Folien und Übungsblätter | | | | | |

| | | | | | |
|--|--|------------|--------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Statistik und Wahrscheinlichkeitsrechnung | | Liste - | Modulnummer 4210 | Sem. 4 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Seminar (2 SWS) | | Häufigkeit jedes Jahr | Bewertung • Klausur oder mündl. Prüfung (80% als PL) • Praktische Tätigkeit als sep. SL (20% als SL) | |
| Aufwand • 60 h Anwesenheit (Vorlesung und Übungen) • 30 h Vor- und Nachbereitung (Vorlesung und Übungen) • 60 h Bearbeitung (Übungsaufgaben) | | | Dozenten Geib, Reith | | Verantwortlich Geib |
| Inhaltliche Voraussetzungen • Analysis: Differential- und Integralrechnung, Untersuchung von Funktionen, Grenzwerte und Folgen, Reihen. • Lineare Algebra: Lineare Gleichungssysteme, Matrizenrechnung | | | | | |
| Lernziele Fähigkeit, elementare stochastische Probleme mit geeigneten Methoden zu lösen • Verständnis wahrscheinlichkeitstheoretischer Grundbegriffe • Kenntnis wichtiger Verteilungen und ihrer Bedeutung • Sicherer Umgang mit den Grundbegriffen der Statistik • Kenntnis grundlegender statistischer Methoden • Beherrschen der wichtigsten Schätz- und Testverfahren | | | | | |
| Inhalte • Kombinatorische Grundlagen und Grundbegriffe der Wahrscheinlichkeitsrechnung • Wahrscheinlichkeit: Definition und Eigenschaften, unabhängige und abhängige Ereignisse • Wahrscheinlichkeitsverteilung einer Zufallsvariablen • Kennwerte oder Maßzahlen verschiedener diskreter Wahrscheinlichkeitsverteilung • Kennwerte oder Maßzahlen einer stetigen Zufallsvariablen (Normalverteilung) • Wahrscheinlichkeitsverteilungen von mehreren Zufallsvariablen • Grundlagen der mathematischen Statistik • Lineare Korrelation und lineare Regression • Statistische Schätzmethoden für die unbekannt Parameter einer Wahrscheinlichkeitsverteilung (Parameterschätzung) • Konfidenzintervalle • Testverfahren | | | | | |
| Literatur Thomas Sonar: Angewandte Mathematik, Modellbildung und Informatik Gerald Teschl, Susanne Teschl: Mathematik für Informatiker, Band 2, Springer, 2006. B. W. Gnedenko: Lehrbuch der Wahrscheinlichkeitstheorie, Verlag Harri Deutsch W. M. Hines, D. C. Montgomery: Probability and Statistics in Engineering and Management Science, John Wiley & Sons Lothar Papula: Mathematik für Ingenieure und Naturwissenschaftler, Band 3, Viewegs, 2001 | | | | | |
| Medienform / Unterrichtssprache • Skript • Übungsblätter und Lösungsblätter auf der Internetseite • Aufgabensammlung mit Lösungen für Klausurvorbereitung | | | | | |

| | | | | | |
|---|--|------------|--------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Management von IT-Prozessen | | Liste - | Modulnummer 4310 | Sem. 4 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Übung (2 SWS) | | Häufigkeit jedes Jahr | Bewertung • Klausur oder mündl. Prüfung (100% als PL) | |
| Aufwand • 60 h Bearbeitung (Übungsaufgaben) • 30 h Vor- und Nachbereitung (Vorlesung und Übungen) • 60 h Anwesenheit (Vorlesung und Übungen) | | | Dozenten Voelz | | Verantwortlich Voelz |
| Inhaltliche Voraussetzungen | | | | | |
| Lernziele Die Studierenden • kennen die grundlegenden Theorien des Prozessmanagement in der Informationstechnik, und • sind in der Lage, deren grundlegende Modelle anzuwenden. • Sie sind in der Lage, eigenständig Gestaltungsentscheidungen zur Produktentwicklung zu fällen, • beherrschen grundlegende Techniken des IT-Projektmanagements, • und kennen verbreitete Vorgehensmodelle und Standards der Softwareentwicklung. | | | | | |
| Inhalte • Grundlagen des Prozessmanagements in der Informationstechnik • Management von Produktentwicklungsprozessen • Projektmanagement in der Informationstechnik • Vorgehensmodelle in der Softwareentwicklung • Normen, Standards und Reifegrade von Softwareprozessen | | | | | |
| Literatur Schmelzer, Sesselmann: Geschäftsprozessmanagement in der Praxis Allweyer: BPMN - Business Process Modeling Notation Freud: Praxishandbuch BPMN Gadatsch: Grundkurs Geschäftsprozess-Management | | | | | |
| Medienform / Unterrichtssprache • Web-Seite zur Veranstaltung • Skript/Folien Übungsblätter als PDF im Netz | | | | | |

| | | | | | |
|---|--|------------|---|---|--|
| Modulbezeichnung / Prüfungsfach Wahlprojekt | | Liste - | Modulnummer 5110 | Sem. 5 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 15 cp, 450 h 8 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (6 SWS) | | Häufigkeit jedes Jahr | Bewertung • Projekt/Fachgespräch (100% als PL) | |
| Aufwand • 90 h Anwesenheit (Projekt, insb. Koordination) • 90 h Literaturstudium, Einarbeitung (Projektinhalte) • 210 h Praktische Tätigkeit (Projekt) • 30 h Anwesenheit (Integriertes Softskill-Seminar) • 30 h Vor- und Nachbereitung (Integriertes Softskill-Seminar) | | | Dozenten Dozenten des Studienbereichs | | Verantwortlich Iglter |
| Inhaltliche Voraussetzungen | | | | | |
| Lernziele Die Fähigkeit zur Auswahl, Bewertung und Anwendung von Konzepten und Methoden zur systematischen (Weiter-)Entwicklung (großer) Softwaresysteme, insbesondere bzgl. phasenübergreifender Querschnittsapekte und der Behandlung der späteren Phasen des Software-Lifecycles, ist für eine verantwortungsvolle Tätigkeit im IT-Bereich jenseits der reinen Programmierung unverzichtbar. Dabei spielt neben guten technischen Kenntnissen auch die Fähigkeit zur koordinierten, arbeitsteiligen Zusammenarbeit in einem Team eine wichtige Rolle. • Management von Softwareprojekten • Organisation des persönlichen Arbeitsprozesses • Software-Tests und Software-Qualitätssicherung • Ganzheitliche Betrachtung des Software-Lebenszyklus incl. Wartung/Pflege/Re-Engineering • Praktische Umsetzung der in Softwaretechnik und dieser Veranstaltung erlernten Konzepte und Methoden in einer umfangreichen Projektaufgabe (im Team zu bearbeiten) Die erworbenen Fähigkeiten tragen in besonderem Maße zur Ausprägung von Projektmanagement- sowie sozialen und Selbst-Kompetenzen bei. Sie erweitern die spezifischen Analyse-, Design- und Realisierungskompetenzen sowie ausgewählte technologische Kompetenzen. | | | | | |
| Inhalte Selbständige Bearbeitung eines größeren Softwareprojekts im Team • Rollenverteilung • Erstellung eines Projektplans • Dokumentation der Projektphasen • Projekt-Controlling • Arbeitsorganisation im Team Methodische Projektbegleitung • Software-Projektmanagement, Projektorganisation • Zeitmanagement, Modelle und Techniken • Umgang mit persönlichen Ressourcen • Arbeiten im Team; Konfliktmanagement • Metriken und Aufwandsschätzung (z.B. Function Point Analysis) • Konfigurations- und Änderungsmanagement (Versionierung, Konfiguration; Toolunterstützung), Software-Qualität (Ziele nach ISO 9126; ISO 9000, CMM) • Testen von Software (Fehlerarten; statische und dynamische Testverfahren; Testdokumentation) • Pflege und Wartung, Umgang mit Software-Altlasten (Legacy Systems); Software-Re-Engineering | | | | | |
| Literatur Siedersleben (Hrsg.): Softwaretechnik, Hanser, 2003 Sommerville: Software Engineering, Addison-Wesley, 2004 Balzert: Lehrbuch der Softwaretechnik, Band II, Spektrum-Verlag, 2000 Bruegge, Dutoit: Object-Oriented Software Engineering, Pearson Studium, 2004 Litke, H.-D.: DV-Projektmanagement - Zeit und Kosten richtig einschätzen. Hanser, neueste Auflage | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Folien, Übungsblätter • Tafel, Flipchart | | | | | |

| | | | | | |
|---|---------------------------------|------------|---|---|--|
| Modulbezeichnung / Prüfungsfach Fachseminar BSc | | Liste - | Modulnummer 5310 | Sem. 5 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 2 SWS | Lehrformen • Seminar (2 SWS) | | Häufigkeit jedes Jahr | Bewertung • Projekt/Fachgespräch (100% als PL) | |
| Aufwand • 30 h Anwesenheit (Seminar) • 60 h Literaturstudium, Einarbeitung (Fachthema) • 60 h Ausarbeitung (Referat) | | | Dozenten Dozenten des Studienbereichs, Gastprofessoren | | Verantwortlich Gergeleit |
| Inhaltliche Voraussetzungen | | | | | |
| Lernziele Selbstständiges Erarbeiten von Inhalten sowie deren verständliche Darstellung von fachlichen Themen sind für den beruflichen Alltag unabdingbar. Absolventen des Moduls • beherrschen selbstständiges Erarbeiten eines Fachthemas anhand von Fachliteratur und anderen Quellen • können ein Fachthema verständlich darstellen und präsentieren • erwerben kommunikative Kompetenz • vertiefen informatische Fachkompetenzen auf dem ausgewählten Themengebiet des Seminars | | | | | |
| Inhalte • Selbstorganisation und selbstständiges Erarbeiten eines Fachthemas • Wissensmanagement und Literaturstudium • Fachliches Schreiben für die schriftliche Ausarbeitung • Präsentationstechnik und Rhetorik für die Präsentation des Themas • Diskussion im Rahmen der Seminarteilnehmer und betreuenden Dozenten | | | | | |
| Literatur Publikationen zum gewählten Themengebiet | | | | | |
| Medienform / Unterrichtssprache • Seminaristischer Unterricht • Präsentationsunterlagen und technische Hilfsmittel • Zusammenfassung der schriftlichen Ausarbeitungen | | | | | |

| | | | | | |
|--|---------------------------------|------------|-------------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Recht für Informatiker | | Liste - | Modulnummer 5320 | Sem. 5 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Seminar (4 SWS) | | Häufigkeit jedes Jahr | Bewertung • Klausur oder mündl. Prüfung (100% als PL) | |
| Aufwand • 60 h Anwesenheit (Sem. Unterricht) • 60 h Vor- und Nachbereitung • 30 h Bearbeitung (Übungsaufgaben) | | | Dozenten Erhard, Lehrbeauftragte | | Verantwortlich Jarass |
| Inhaltliche Voraussetzungen | | | | | |
| Lernziele Der Studierende beherrscht die Grundzusammenhänge des Privatrechtes sowie das problembewusste Erkennen von zivilrechtlichen praxisbezogenen Grundfällen im Arbeitsumfeld eines Informatikers. | | | | | |
| Inhalte Es werden die wesentlichen Grundzüge des Privatrechts und die relevanten Rechtsbegriffe erörtert. Hierbei werden für den Informatiker wesentliche grundsätzliche rechtliche Zusammenhänge wie Privatautonomie, Zwingendes Recht, Abstraktionsprinzip, Recht der Kaufleute etc. vermittelt. Darüber hinaus wird die Arbeitsweise des Juristen an praxisorientierten Fällen für den Informatiker dargestellt. Vertragsrechtliche Grundzüge, Grundlagen der Sachmängelhaftung und des Schadensersatzrechtes, Grundzüge des Internet- und des Datenschutzrechtes sowie die sich aus der Informatik ergebenden rechtlichen Besonderheiten werden in einem Vertiefungsteil behandelt. | | | | | |
| Literatur Skript | | | | | |
| Medienform / Unterrichtssprache • Skript • Folien • Beamer • Tafel | | | | | |

| | | | | | |
|--|-----------------------------------|------------|-----------------------|-----------|--|
| Modulbezeichnung / Prüfungsfach Praxisprojekt | | Liste - | Modulnummer 6100 | Sem. 6 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 15 cp, 450 h 0 SWS | Lehrformen • Praktikum (0 SWS) | | Häufigkeit ständig | Bewertung | |
| Aufwand • 30 h Vor- und Nachbereitung (Projektbericht) • 420 h Projektarbeit (im Unternehmen) | | | Dozenten | | Verantwortlich Geib |
| Inhaltliche Voraussetzungen | | | | | |
| <p>Lernziele</p> <p>Das Berufspraktikum bietet den Studierenden die Möglichkeit, ihre in mehreren Semestern an der Hochschule erworbenen fachlichen Fähigkeiten in der Praxis zu erproben und zusätzlich wichtige Kompetenzen im außerfachlichen Bereich zu erwerben. Es spielt daher im Rahmen einer praxisorientierten und arbeitsmarktgerechten Ausbildung sowie zur Persönlichkeitsbildung eine zentrale Rolle.</p> <ul style="list-style-type: none"> • Orientierung im angestrebten Berufsfeld • Erwerb praktischer Kenntnisse und kennen lernen berufstypischer Arbeitsweisen • Kennen lernen technischer und organisatorischer Zusammenhänge, die für das Berufsfeld typisch sind • Beteiligung am Arbeitsprozess entsprechend dem Ausbildungsstand • Praktische Ausbildung an fest umrissenen, konkreten Projekten | | | | | |
| <p>Inhalte</p> <p>Praktische Tätigkeit mit deutlichem Informatik-Schwerpunkt, z.B.</p> <ul style="list-style-type: none"> • Durchführung von Anforderungsanalysen, Erstellung Softwareentwurf • Programmierung • Datenbankentwurf und -implementierung • Realisierung von Web-Anwendungen • Netzwerkplanung, Sicherheitsanalysen • Verarbeitung von Graphikdaten, Visualisierung <p>Rahmenbedingungen:</p> <ul style="list-style-type: none"> • Kürzere tägliche Arbeitszeiten als ein halber Tag sind nicht erlaubt. • Das gesamte Praktikum ist innerhalb von 9 Monaten abzuleisten. • Krankheits- und andere Ausfallzeiten zählen dabei nicht mit. • Das Berufspraktikum muss bei einer Praktikumsstelle absolviert werden. • Über das Berufspraktikum ist ein Bericht von mindestens 17 Seiten anzufertigen. • Es gibt nach Abschluss des Praktikums ein Fachgespräch mit dem hochschulseitigen Betreuer über die Praktikumsinhalte. | | | | | |
| <p>Literatur</p> <p>keine</p> | | | | | |
| <p>Medienform / Unterrichtssprache</p> <ul style="list-style-type: none"> • Tätigkeits- oder Projektbericht (gedruckt, ca. 20 DIN A4) | | | | | |

| | | | | | |
|---|---|------------|--|--|--|
| Modulbezeichnung / Prüfungsfach Bachelor-Thesis (mit Kolloquium) | | Liste - | Modulnummer 9040 | Sem. 6 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 15 cp, 450 h 0 SWS | Lehrformen • Bachelor-Arbeit (0 SWS) • Kolloquium (0 SWS) | | Häufigkeit ständig | Bewertung • Bachelor-Arbeit (80% als PL) • Kolloquium (20% als PL) | |
| Aufwand • 90 h Kolloquium (Vorbereitung und Anwesenheit) • 360 h Abschlussarbeit (selbstständige, betreute Durchführung; Erstellung Bachelor-Arbeit) | | | Dozenten Dozenten des Studienbereichs | | Verantwortlich Gergeleit |
| Inhaltliche Voraussetzungen | | | | | |
| Lernziele Selbständige praxistaugliche Anwendung der im Studium gelernten Konzepte und Methoden zur Lösung einer begrenzten Aufgabenstellung in begrenzter Zeit im Schwierigkeitsbereich von fortgeschrittenen Lehrbüchern unter Einschluss einiger Bereiche des aktuellsten Wissensstandes des Studienfaches. Durch Erbringen des Moduls werden die folgenden Kompetenzen nachgewiesen: <ul style="list-style-type: none"> • Kompetenz zur Analyse komplexer, praxistauglicher Aufgabenstellungen • Kompetenz zur Bewertung verschiedener Lösungsalternativen • Kompetenz zur Realisierung von Lösungen auf Basis aktueller Technologien • Kompetenz zur Weiterentwicklung von Modellen und Technologien der Informatik im bearbeiteten Themenbereich • Kommunikative Kompetenz durch Präsentation und Verteidigung der eigenen Arbeiten • Projektmanagement-Kompetenzen, insb. Zeit-Management | | | | | |
| Inhalte <ul style="list-style-type: none"> • Analyse der Aufgabenstellung • Erarbeiten der theoretischen Grundlagen, Bewerten verschiedener Lösungsalternativen • Selbständige Entwicklung der Lösung für die Aufgabenstellung • Dokumentation in Form der Bachelor Thesis | | | | | |
| Literatur Literatur passend zum Themengebiet | | | | | |
| Medienform / Unterrichtssprache <ul style="list-style-type: none"> • Bachelor Thesis (gedruckt, gebunden), 4 Exemplare | | | | | |

| | | | | | |
|---|--|-------------|--|--|--|
| Modulbezeichnung / Prüfungsfach 3D-Modellierung & Animation | | Liste IB | Modulnummer 7100 | Sem. 4, 5 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben) | | | Dozenten Dörner, Schulz, Schwanecke | | Verantwortlich Schulz |
| Inhaltliche Voraussetzungen • Objektorientierte Softwareentwicklung: Programmierkenntnisse • Computergrafik: Grundkenntnisse der generativen Computergrafik und über GDV-Systeme | | | | | |
| Lernziele Die Studierenden sollen • Fertigkeiten in der 3D-Modellierung besitzen • Neben traditionellen Modellierungsverfahren auch Erfahrungen mit 3D-Rekonstruktion / Reverse Engineering haben • Software zur Modellierung und Animation (z.B. Maya, 3D Studio Max, Blender) anwenden können • Verschiedene Animationsparadigmen (key frame, physics engine) beherrschen Die erworbenen Fähigkeiten tragen damit in besonderem Maße zur Ausprägung von spezifischen Analyse-, Design- und Realisierungskompetenzen und zur Erweiterung von spezifischen technologischen Kompetenzen bei. | | | | | |
| Inhalte • Akquisition von 3D-Objektmodellen und Szenenmodellen • Videobasierte 3D-Rekonstruktion • 3D-Scanning • Photogrammetrie • Terraingenerierung • Reverse Engineering • Objektrepräsentationen (NURBS, Punktwolken, Volumetrische Repräsentationen, Hybride Repräsentationen) • Software zur Modellierung und Animation (z.B. Maya, 3D Studio Max) • Keyframe-Animation • Physik-Animation • Motion Capturing • Partikelsysteme | | | | | |
| Literatur Hoschek, Lasser: Grundlagen der Geometrische Datenverarbeitung, Teubner Farin: Curves and Surfaces for CAGD, Morgan Kaufmann Publishers, 2001 Kerlow: The Art of 3D Computer Animation and Effects, John Wiley + Sons, 4th Ed., 2009 Ausgewählte Originalliteratur | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter | | | | | |

| | | | | | |
|---|--|-------------|--|--|--|
| Modulbezeichnung / Prüfungsfach Ausgewählte Kapitel der Angewandten Informatik | | Liste IB | Modulnummer 7110 | Sem. 4, 5 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 60 h Bearbeitung (Praktikumsaufgaben) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Anwesenheit (Vorlesung und Praktikum) | | | Dozenten Dozenten des Studienbereichs, Gastprofessoren, Lehrbeauftragte | | Verantwortlich Gergeleit |
| Inhaltliche Voraussetzungen | | | | | |
| Lernziele Erarbeitung von aktuellen/wechselnden Themen aus der angewandten Informatik. • Nutzen vorhandener Kenntnisse, um neue Themengebiete zu erschließen. • Umsetzung von Anwendungsszenarien unter Einsatz aktueller Konzepte und Technologien. | | | | | |
| Inhalte • Veranstaltungsinhalte in Abhängigkeit vom gewählten Thema | | | | | |
| Literatur Aktuelle Lehrbücher und Papiere zum gewählten Thema | | | | | |
| Medienform / Unterrichtssprache • Skript/Folien und Übungsblätter | | | | | |

| | | | | | |
|--|--|-------------|--------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Compilerbau | | Liste IB | Modulnummer 7120 | Sem. 4, 5 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 60 h Bearbeitung (Praktikumsaufgaben) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Anwesenheit (Vorlesung und Praktikum) | | | Dozenten Panitz | | Verantwortlich Panitz |
| Inhaltliche Voraussetzungen | | | | | |
| Lernziele Elemente des Compilerbau gehören zu den typischen Aufgaben von Informatikern: <ul style="list-style-type: none"> • Compiler für kleinere Sprachen entwerfen und implementieren • Scanner und Parser für Kommandosprachen programmieren und einsetzen • Compilergeneratoren verwenden • Grammatiken analysieren, bewerten und transformieren | | | | | |
| Inhalte *Einführung (Definition eines Compilers, Analyse-Synthese-Modell, Phasen, Umgebung eines Compilers, Beispiele zur Compilation) <ul style="list-style-type: none"> • Sprachanalyse (Formale Sprachen, Grammatiken, Chomsky-Hierarchie, Eigenschaften von Grammatiken, Darstellung von Produktionen, eine Modellsprache, Semantik von Programmiersprachen) • Lexikalische Analyse (Scanner, Implementierungsmöglichkeiten, Scanner der Modellsprache als Beispiel, Scannergeneratoren) • Syntaktische Analyse (Top-Down-Analyse, LL(1)-Grammatiken, Rekursiver Abstieg, Bottom-Up-Analyse, Parser-Generatoren, Parser für die Modellsprache als Beispiel) • Semantische Analyse (Definition und Überblick, Typ-Prüfung, Gültigkeitsprüfung, Vorgehensweise bei der Modellsprache) • Fehlerbehandlung (Vorgehensweise, Fehlerbehandlung bei Syntaxanalyse) • Laufzeit-Speicherverwaltung (Grundlagen, Adressierung, Aufteilung des Laufzeitspeichers, Activation Records, Dynamic-Link- und Static-Link-Ketten) • Code- und Zwischencode-Generierung (Syntaxisorientierte Übersetzung, Zwischensprachen, Semantische Aktionen, Zwischensprache des Modell-Compilers, Code-Erzeugung, Interpretation, Assemblercode-Erzeugung, Prinzipien der Optimierung) • Compiler-Erweiterung und Portierung (T-Diagramme, Erweiterung, Bootstrap, Portierung) | | | | | |
| Literatur Wirth, N.: Grundlagen und Techniken des Compilerbaus Sethi, R.; Lam, M.; Aho, A.: Compiler. Prinzipien, Techniken und Tools Bauer, B.; Höllerer, R.: Übersetzung objektorientierter Programmiersprachen | | | | | |
| Medienform / Unterrichtssprache <ul style="list-style-type: none"> • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter als PDF-Files | | | | | |

| | | | | | |
|--|--|-------------|--|--|--|
| Modulbezeichnung / Prüfungsfach Computergrafik für Education & Entertainment | | Liste IB | Modulnummer 7130 | Sem. 4, 5 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben) | | | Dozenten Dörner, Schulz, Schwanecke | | Verantwortlich Dörner |
| Inhaltliche Voraussetzungen • Objektorientierte Softwareentwicklung: Programmierkenntnisse • Computergrafik: Grundkenntnisse der generativen Computergrafik und über GDV-Systeme | | | | | |
| Lernziele E-Learning und Entertainment sind wichtige Anwendungen, für die Informatiker Softwaresysteme unter Verwendung von Computergrafik konzipieren und realisieren. Ziel ist es, dass die Studenten Grundlagen für eine erfolgreiche Gestaltung von derartigen Anwendungen kennen und Software für spezielle Aufgabenstellungen (unter Beachtung der anwendungstypischen Randbedingungen und Produktionsprozessen) entwickeln können. Dabei sollen die Studenten auch Kompetenzen im Umgang mit dedizierten Autorenwerkzeugen, Skriptingssprachen und Beschreibungssprachen erwerben. | | | | | |
| Inhalte • e-Learning – Stärken und Schwächen • Grundlagen aus Pädagogik und Didaktik • Digital Storytelling und Game Design • Simulierte Welten, Edutainment und Serious Games • Kollaborative Lernumgebungen – Multiplayer Games • Autorensysteme (z.B. Flash, Quest3D) • Einsatz von Game Engines • Produktionsprozesse • Lernplattformen und Standards (z.B. SCORM) | | | | | |
| Literatur U. Riser, J. Keuneke, H. Freibichler, B. Hoffmann: Konzeption und Entwicklung interaktiver Lernprogramme, Springer, 2002 M. Wendt: CBT und WBT – konzipieren, entwickeln, gestalten, Hanser, 2003 S. Rabin: Introduction to Game Development, Cengage Learning Services, 2009 | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Aufgabenblätter (als pdf-Dateien) • e-Learning Module zur selbständigen Ergänzung für das Praktikum | | | | | |

| | | | | | |
|--|--|-------------|----------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Concurrent Programming | | Liste IB | Modulnummer 7140 | Sem. 4, 5 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 60 h Bearbeitung (Praktikumsaufgaben) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Anwesenheit (Vorlesung und Praktikum) | | | Dozenten Barth, Kröger, Reith | | Verantwortlich Kröger |
| Inhaltliche Voraussetzungen • Modul Programmiermethodik und Programmieretechniken: Programmieren in Java, C, C++ • Modul Betriebssysteme: Konzept der Prozesse und Threads | | | | | |
| Lernziele Nach erfolgreichem Abschluss der Veranstaltung • sind die Studierenden vertraut mit den Möglichkeiten und Grenzen nebenläufiger Verarbeitung auf Hardware- und Software-Ebene • kennen sie bekannte Synchronisations- und Kommunikationsmuster und können diese bei der Entwicklung nebenläufiger Anwendungen einsetzen • sind sie in der Lage, parallele Algorithmen in verschiedenen Programmiermodellen umzusetzen Das Modul erweitert die technologischen Kompetenzen in Hinblick auf Parallele Hardware-Architekturen, vertieft die algorithmischen Kompetenzen zum systematischen Umgang mit Nebenläufigkeit und festigt die Design- und Realisierungskompetenzen durch die Bearbeitung der Praktikumsaufgaben. | | | | | |
| Inhalte • Maschinenmodelle für nebenläufige und parallele Verarbeitung, Multicore-Prozessoren, Distributed Shared Memory • Programmiermodelle • Klassische Synchronisationsprobleme • Abstraktionen für Synchronisation und Kommunikation • Typische Synchronisations- und Kommunikationsmuster • Neuere Ansätze (z.B. Transactional Memory) • Bewertung und Performance-Messungen • POSIX Pthreads • Java Concurrency Framework java.util.concurrent • C++ Boost Libraries und Concurrency Framework des C++ Standards • Actor-Modell und Scala • OpenMP • Programmierung von Graphikkarten (GPGPUs) | | | | | |
| Literatur Rauber, Rürger: Multicore: Parallele Programmierung, Springer 2008 Goetz, Peierls, Bloch, Bowbeer, Holmes, Lea: Java Concurrency in Practice, Addison-Wesley, 2006 Williams: C++ Concurrency in Action, Addison-Wesley, 2009 | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter als PDF | | | | | |

| | | | | | |
|--|--|-------------|--------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Datenbank-Technologien | | Liste IB | Modulnummer 7150 | Sem. 4, 5 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 60 h Bearbeitung (Praktikumsaufgaben) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Anwesenheit (Vorlesung und Praktikum) | | | Dozenten Muth | | Verantwortlich Muth |
| Inhaltliche Voraussetzungen • Datenbanken | | | | | |
| Lernziele Datenbanksysteme werden von jedem Informatiker eingesetzt. Für bestimmte Anwendungsfälle ist das Wissen um die Interna von Datenbanksystemen notwendig oder die Erweiterung von Datenbanksystemen für spezifische Einsatzszenarien. • Datenbankverwendung mit Wissen der internen Strukturen optimieren. • Datenbanksysteme für spezifische Anwendungsszenarien erweitern. | | | | | |
| Inhalte • Datenbank-Implementierungstechniken für relationale Datenbanken (z.B. Indexstrukturen, Puffer, Hintergrundspeicher, Anfrageoptimierung, Transaktionsverwaltung, Recovery und Sicherung) • Techniken und Verfahren für spezielle Anwendungen (z.B. Zugriffsstrukturen für Geometrische/Multimedia/ Text Daten, Objektdatenbanken und hierarchische Daten, OLAP, Verteilte Datenbanken) | | | | | |
| Literatur Härder, Rahm: Datenbanksysteme. Konzepte und Techniken der Implementierung, Springer, 2001 Marco: Building and Managing the Meta Data Repository, Wiley, 2000 Kemper, Eickler: Datenbanksysteme. Eine Einführung, Oldenbourg, 2009 Saake, Heuer, Sattler: Datenbank-Implementierungstechniken, Mitp-Verlag, 2005 Vossen: Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme, Oldenbourg, 2008 | | | | | |
| Medienform / Unterrichtssprache • Web_Seite zur Veranstaltung • Skript/Folien und Übungsblätter als PDF | | | | | |

| | | | | | |
|---|--|-------------|--|--|--|
| Modulbezeichnung / Prüfungsfach Digitale Bildverarbeitung | | Liste IB | Modulnummer 7160 | Sem. 4, 5 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben) | | | Dozenten Dörner, Schulz, Schwanecke | | Verantwortlich Schwanecke |
| Inhaltliche Voraussetzungen • Objektorientierte Softwareentwicklung: Programmierkenntnisse • Computergrafik: Grundkenntnisse der generativen Computergrafik und über GDV-Systeme | | | | | |
| Lernziele Die Studierenden sollen befähigt werden • die Arbeitsweisen der digitalen Bildanalyse zu verstehen, • standardisierte Analyseverfahren auf Bilder anzuwenden, • interaktionsarme Verfahren auf der Basis von a-priori-Wissen zu entwickeln | | | | | |
| Inhalte • Eigenschaften diskretisierter Bilder • Grundlagen der Videotechnik • Analyse-Tools zur Bewertung digitaler Bilder • Lineare und nichtlineare Filterungen • Strukturanalyse • Lineare und nichtlineare Bildverzerrungen • Transformationen von Hough und Radon • Bildgebende Verfahren für spezielle Anwendungen • Farbmimetrik • Orthogonale Basistransformationen • Bilddatenkomprimierung • Morphologie | | | | | |
| Literatur Haberecker: Digitale Bildverarbeitung, Grundlagen und Anwendungen, Carl Hanser Verlag, 1991 Jähne: Digitale Bildverarbeitung, Springer, 2005 Gonzalez, Wintz: Digital Image Processing, Longman, 1987 Burger, Burge: Digitale Bildverarbeitung: Eine Einführung mit Java und ImageJ, Springer, 2008 | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter (als pdf-Dateien) | | | | | |

| | | | | | |
|--|--|-------------|--------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Echtzeitverarbeitung | | Liste IB | Modulnummer 7170 | Sem. 4, 5 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben) | | | Dozenten Kaiser, Kröger | | Verantwortlich Kaiser |
| Inhaltliche Voraussetzungen • BS+RA: Grundlagen von Betriebssystemen | | | | | |
| Lernziele Verarbeitung von Information unter Echtzeitbedingungen hat über die klassischen technischen Anwendungen hinaus auch in vielen anderen Anwendungsbereichen Einzug gehalten. Die Basis für die Möglichkeiten wird auf der Betriebssystemebene gelegt. Nach erfolgreichem Abschluss der Lehrveranstaltung • kennen die Studierenden Struktur, Schnittstellen und Kenngrößen von Echtzeitbetriebssystemen und können diese beurteilen • kennen sie Anforderungen von Echtzeitanwendungen und können diese beurteilen • sind sie befähigt, die Programmierung von Echtzeitanwendungen durchführen zu können • können sie experimentell Kenngrößen von Echtzeitsystemen bestimmen. Insgesamt trägt das Modul zu einer Erweiterung der technologischen Kompetenzen im Bereich der technischen Informatik bei und vertieft die Design- und Realisierungskompetenzen bzgl. der Entwicklung systemnaher Software. | | | | | |
| Inhalte • Einführung • Grundlagen (Grundbegriffe, nicht-funktionale Eigenschaften, Vorhersagbarkeit, Verlässlichkeit, Klassifizierung von Echtzeitsystemen, Wertfunktionen, Beispiele) • I/O-Programmierung und Treiber • Architektur von Echtzeitbetriebssystemen (Klassifizierung, Standards, grundlegende BS-Kernabläufe, BS-Organisationsformen) • Beispiele (POSIX, Linux mit Echtzeiterweiterungen, uCOS-II, OSEK) • Planungsverfahren / Scheduling (Modellbildung, Planen durch Suchen, Planen nach Fristen, Spielräumen, monotonen Raten, Bewertung und Vergleich, Planen und Synchronisation) • Spezielle Aspekte der Softwaretechnik (Software-Qualität, Modellierung UML/SysML, UML MARTE, Funktionale Sicherheit, Validierung, systematisches Testen, Leistungsbewertung) • Echtzeitprogrammierung (Merkmale, Echtzeitsprachen und ihre Programmiermodelle, MISRA C, C++, Ada) • Einführung in die Regelungstechnik für Informatiker • Verteilte Echtzeitsysteme (Feldbusse, Ethernet und Echtzeit, Wireless-Netze und Echtzeit, globale Zeit und Uhrensynchronisation, Gruppenkommunikation und Echtzeit) • Echtzeit-Middleware (z.B. RT CORBA, DDS, CANopen, OPC UA) • Praktikum (Cross-Entwicklungsumgebungen [Beispiele: Linux, Embedded PCs, ARM], Interrupt-Handler, Treiber, Echtzeitprogrammierung [Beispiele: uCOS-II, Ada], Echtzeitprogr. mit Pthreads, experimentelle Bestimmung von Kenngrößen eines RTOS, systematisches Testen, kleine Anwendungsexperimente) | | | | | |
| Literatur Zöbel, D.: Echtzeitsysteme - Grundlagen der Planung, Springer, 2008 Wörn, H.; Brinkschulte, U.: Echtzeitsysteme - Grundlagen, Funktionsweisen, Anwendungen, Springer, 2005 Gallmeister, B.O.: POSIX.4: Programming for the Real World, O'Reilly & Associates, 1995 Buttazzo: Hard Real-Time Computing Systems, Kluwer, 1997 | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Folien und Übungsblätter als pdf | | | | | |

| | | | | | |
|---|--|-------------|--------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Fehlertolerante Systeme | | Liste IB | Modulnummer 7180 | Sem. 4, 5 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben) | | | Dozenten Geib | Verantwortlich Geib | |
| Inhaltliche Voraussetzungen • Techn. Grundlagen der Informatik: Codierung und Schaltungstechnik | | | | | |
| Lernziele Die Studierenden werden in die Lage versetzt, Fehlerursachen und Fehlerauswirkungen analysieren und geeignete Gegenmaßnahmen zur Erzielung einer vorgegebenen System-Verfügbarkeit oder System-Lebensdauer anwenden zu können. Dazu zählt u. a.: • Die Aneignung der Grundlagen zur Klassifizierung, Beschreibung und Analyse von zuverlässigkeitstechnischen Problemstellungen. • Das Kennenlernen der grundlegenden Methoden, die zur Beurteilung von Redundanzeigenschaften sowie der Fehlertoleranz herangezogen werden. • Die Auswahl und Beurteilung von Fehlertoleranzmaßnahmen, die ein System weniger fehleranfällig machen gegenüber äußeren Einflüssen sowie gegen inhärente Schwachstellen. | | | | | |
| Inhalte Strukturierter Einstieg in das Thema Funktionssicherheit, Zuverlässigkeit und fehlertolerante Systeme: • Einführung in die Thematik (Begriffe und Benennungen, Aufgaben und Zweck, Kenngrößen und Beschreibungsmittel, Anwendungsbereiche) • Grundlagen fehlertoleranter Rechensysteme (Fehlerursachen und Fehlerauswirkungen, Fehlerarten, Risikostufen und Kritikalität) • Graphische Hilfsmittel und systemtheoretische Grundlagen (Redundanzstrukturfunktion, Fehlerbäume, Zuverlässigkeitsblockdiagramme, Zustandsdiagramme und Petrinetze, Zuverlässigkeitskenngrößen reparierbarer und nichtreparierbarer Systeme) • Mathematische Behandlung von Zuverlässigkeitsproblemen (Verfügbarkeit einfacher und vermaschter Systemstrukturen wie Serien- und Parallelredundanz, m-von-n-Systeme, Serien-Parallel- und Parallel-Serien-Systeme) • Darstellung von Fehlertoleranzverfahren (Komponentenvervielfachung, fehlerkorrigierende Codes, Fehlermaskierung und Mehrheitsentscheidung, Fehlerisolierung und Fehlerentkopplung) • Implementierung von Fehlertoleranzverfahren (HW- und SW-implementierte Fehlertoleranzmaßnahmen, Kryptosysteme, Diversität) | | | | | |
| Literatur Görke, W.: Fehlertolerante Rechensysteme, Oldenbourg Verlag (1989) Schneeweiss, W. G.: Zuverlässigkeitstechnik - von den Komponenten zum System, Datakontext-Verlag (1992) Biolini, A.: Qualität und Zuverlässigkeit technischer Systeme, Springer-Verlag (1985) Störmer, H.: Mathematische Theorie der Zuverlässigkeit elektronischer Systeme, Oldenbourg Verlag (1970) | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter als PDF | | | | | |

| | | | | | |
|--|--|-------------|--------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Funktionale Programmierung | | Liste IB | Modulnummer 7190 | Sem. 4, 5 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 60 h Bearbeitung (Praktikumsaufgaben) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Anwesenheit (Vorlesung und Praktikum) | | | Dozenten Panitz | | Verantwortlich Panitz |
| Inhaltliche Voraussetzungen | | | | | |
| Lernziele Programme als mathematische Funktionen ohne Seiteneffekte verstehen. Semantik von Programmen erfassen und über Programme formal mathematisch schließen können und Aussagen treffen. | | | | | |
| Inhalte <ul style="list-style-type: none"> • Milner Typsystem, Typinferenz • Programme höherer Ordnung • Pattern Matching • List Comprehensions • Typklassen • Polymorphisches Programmieren, Metaprogramme • verzögerte und strikte Auswertung • Monaden als Weg der Integration imperativer Konzepte • dynamische Typen • Modelle der Nebenläufigkeit, Aktoren • semantische Grundlagen über den Lambda-Kalkül • Verifikationstechniken Zum Einsatz kommende Programmiersprachen u.a. können sein: ML, F#, Scala, Haskell, Clean, Erlang | | | | | |
| Literatur Richard Bird: Introduction to Functional Programming using Haskell, Prentice Hall, 2nd ed., 1998. Martin Erwig: Grundlagen funktionaler Programmierung, Oldenbourg-Verlag, 1999. Martin Odersky, Lex Spoon, Bill Venner: Programming in Scala - A comprehensive step-by-step guide, Artima Inc, 2008 David Pollak: Beginning Scala, Apress, 2009 | | | | | |
| Medienform / Unterrichtssprache <ul style="list-style-type: none"> • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter als PDF-Files | | | | | |

| | | | | | |
|--|--|-------------|--------------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Hardware/Software-Schnittstellen | | Liste IB | Modulnummer 7200 | Sem. 4, 5 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben) | | | Dozenten Gergeleit, Grothe, Thoss | | Verantwortlich Gergeleit |
| Inhaltliche Voraussetzungen • TechInfo: Grundlagen Digitaltechnik und Rechnerorganisation • RNT: Bitübertragung und Modulationsverfahren • BSVS: Gerätetreiber • Programmieren 1: Imperatives Programmieren, Umgang mit Programmierungswerkzeugen | | | | | |
| Lernziele Die Interaktion insbesondere von Rechensystemen mit ihrer Umwelt basiert auf dem Zusammenspiel von Hardware-Schnittstellen und deren Bedienung durch Software. Für die erfolgreiche Konstruktion von eingebetteten Systemen und Peripherielementen sind Kenntnisse der Modellierung und Implementierung dieser hardwarenahen Ebene unterhalb der Treiberschichten von Betriebssystemen unabdingbar. Dieses Modul befähigt die Studierenden: • Datenblätter und Applikationsschaltungsbeispiele von Hardware-Herstellern zu verstehen • für eine Problemstellung geeignete Peripherieschaltungen auswählen und anwenden zu können • Mikrocontroller für einen Anwendungsfall passend auswählen und einsetzen zu können • Fehlersuche in Hard- und Software eingebetteter Systeme zu beherrschen • mit Hardware-Entwicklern zusammenzuarbeiten • Softwareentwicklung auf Plattformen mit stark eingeschränkten Hardware-Ressourcen betreiben zu können Die erworbenen Fähigkeiten tragen damit in besonderem Maße zur Ausprägung von Hardware-spezifischen Analyse-, Design- und Realisierungskompetenzen und darüber hinaus zum Erwerb von spezifischen technologischen Kompetenzen bei. | | | | | |
| Inhalte • Grundlagen elektronischer Bauelemente und Schaltungstechnik für das Design von Peripherielementen • Simulation von Elektronikschaltungen • PC-gesteuerte Messgeräte • Integration von Rechensystemen mit ihrer Umwelt • Klassen von Sensoren und Aktoren, Messumformer und physikalische Größen • Systemdesign mit Mikrocontroller-Plattformen • Beachtung von nicht-funktionalen Anforderungen (z.B. Energieeffizienz und Kosteneffizienz) • Integration von Hard- und Softwarekomponenten zu anwendungsspezifischen Systemen • Analogsignale: Verstärkung, Filterung, Wandlung • Leistungselektronik zur Ansteuerung von Lasten • Handwerklich einwandfreies Arbeiten in der Elektronik und Messtechnik Schwerpunkthemen für Projekte • Architektur von Mikrorechenknoten für Ubiquitous Computing und Sensornetze • Programmierbare Hardware (PLDs, FPGAs), Hardware-Beschreibungssprachen (VHDL) • HF-Schaltungsdesign für Funkanwendungen • Selbst entwickelte Laborgeräte für Entwicklung und Fehlersuche • Minimalistische Betriebssysteme (KB-Footprint) • Design und Firmware von USB Devices • Anwendungen mit SoC-Controllern (System on Chip) | | | | | |
| Literatur U.Tietze, Ch.Schenk: Halbleiter-Schaltungstechnik, Springer-Verlag Christian Siemers, Axel Sikora: Taschenbuch Digitaltechnik, Hanser-Verlag Catsoulis: Designing Embedded Hardware, O'Reilly-Verlag T. Beierlein, O. Hagenbruch: Taschenbuch Mikroprozessortechnik, Hanser-Verlag | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript und Aufgabenblätter als pdf-Dateien • Projektbesprechungen mit Anschrieb und Datenprojektor • Lehr- und Datenbücher | | | | | |

| | | | | | |
|--|--|-------------|------------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Künstliche Intelligenz | | Liste IB | Modulnummer 7210 | Sem. 4, 5 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 60 h Bearbeitung (Praktikumsaufgaben) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Anwesenheit (Vorlesung und Praktikum) | | | Dozenten Barth, Krechel, Panitz | | Verantwortlich Krechel |
| Inhaltliche Voraussetzungen | | | | | |
| Lernziele Methoden der künstlichen Intelligenz finden zunehmend breiteren Einsatz bei Datenbankanwendungen, Optimierungsfragestellungen, automatisierten Problemlösungen und in der Automatisierung. • Methoden der künstlichen Intelligenz insbesondere auch im Vergleich zu herkömmlichen Verfahren bewerten und an Beispielen einsetzen • Erfahrung im Einsatz geeigneter Repräsentationen gewinnen • Grundlagen von Maschinenlernen und Bilderkennung anwenden | | | | | |
| Inhalte • Überblick zu Anwendungsgebieten der künstlichen Intelligenz • Wissensrepräsentation und Inferenz, Semantische Netze • Generiere-und-Teste-Verfahren, Suchverfahren • Prädikatenlogik, Regeln und Regelverarbeitung, Wissensmodellierung • Grundlagen zu Prolog • Frames und Frameverarbeitung • Numerische/Symbolische Beschränkung und Propagierung • Natürliche Sprache, Spracherkennung, Syntaktische Analyse, Semantik | | | | | |
| Literatur Luger: Artificial Intelligence, Structures and Strategies for Complex Problem Solving, Addison-Wesely, 2004 Cawsey: Künstliche Intelligenz, Pearson, 2003 Russell, Norvig: Künstliche Intelligenz: Ein moderner Ansatz, Pearson, 2004 Lämmel, Cleve: Künstliche Intelligenz, Fachbuchverlag, 2008 | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter | | | | | |

| | | | | | |
|--|--|-------------|--|--|--|
| Modulbezeichnung / Prüfungsfach Methoden und Anwendungen der Computergrafik | | Liste IB | Modulnummer 7220 | Sem. 4, 5 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben) | | | Dozenten Dörner, Schulz, Schwanecke | | Verantwortlich Schulz |
| Inhaltliche Voraussetzungen • Objektorientierte Softwareentwicklung: Programmierkenntnisse • Computergrafik: Grundkenntnisse der generativen Computergrafik und über GDV-Systeme | | | | | |
| Lernziele Das Gebiet der Computergrafik (Synthese und Analyse) befindet sich nach wie vor in einer sehr dynamischen Entwicklung. Einige Paradigmenwechsel, wie z.B. die Ablösung des lokalen Beleuchtungsmodells durch Echtzeit-Raytracing sind bereits absehbar, andere werden in der mittleren Zukunft hinzukommen. Dieser Modul soll die wichtigen neuen Entwicklungen aufnehmen, so dass die Studierenden • Methoden und Anwendungen kennenlernen, die in naher Zukunft am Markt eingeführt werden • Methoden, Hard- und Software anwenden können, die zur Zeit der Durchführung des Moduls gerade neu am Markt eingeführt wurden | | | | | |
| Inhalte Naturgemäß lassen sich Inhalte hier nur beispielhaft aufzählen, da dieser Modul ja Entwicklungen aufgreifen soll, die zum Zeitpunkt der Modulbeschreibung teilweise noch nicht im Einzelnen absehbar sind. • Globale Beleuchtungsmodelle und Echtzeit-Raytracing • Physikbasierte Animation | | | | | |
| Literatur Ausgewählte Originalliteratur | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter | | | | | |

| | | | | | |
|---|--|-------------|--------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Security | | Liste IB | Modulnummer 7240 | Sem. 4, 5 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben) | | | Dozenten Geib, Reith | | Verantwortlich Geib |
| Inhaltliche Voraussetzungen • Lineare Algebra: Abbildungen, Matrizenrechnung, lineare Gleichungssysteme | | | | | |
| Lernziele Die Studierenden werden an Fallbeispielen in die Lage versetzt, die Ursachen für Securityprobleme in informationstechnischen Systemen analysieren und den Problembereichen grundlegende Sicherheitskonzepte gegenüberstellen zu können. Dazu zählt u. a.: • Das Kennenlernen von technischen und nichttechnischen Gegenmaßnahmen, die erforderlich sind, um die vielfältigen Bedrohungen abzuwehren, denen IT-Systeme heutzutage ausgesetzt sind. • Die Aneignung fundierter Kenntnisse über die Wirksamkeit kryptographischer Verfahren und Protokolle. • Das Herausstellen der methodischen, systematischen Vorgehensweise bei der Konstruktion und Anwendung sicherer Systeme. | | | | | |
| Inhalte Fundierte Einführung in die technischen Grundlagen und Konzepte der heutigen Sicherheitstechnik sowie in das Grundwissen über deren Algorithmen, Protokolle und Verfahren: • Einführung in die IT-Sicherheit (grundlegende Begriffe, Schutzziele, Schwachstellen, Bedrohungen, Angriffe, Sicherheitsstrategien) • Algebraische Strukturen und elementare Zahlentheorie (Restklassen modulo m , Primzahlen und Teiler, Euklidischer Algorithmus und Kongruenzen, Hashing) • Spezielle Bedrohungen (Buffer-Overflows, Computerviren und Trojanische Pferde, Man-in-the-Middle-Attacks, Denial-of-Service Angriffe, Passwort-Crack) • Monoalphabetische Chiffren und deren Analyse (differenzielle und lineare Kryptoanalyse) • Security Engineering (Bedrohungsanalyse, Risikoanalyse, Schutzbedarfsermittlung, Penetrationstests, Sicherheitsstrategien) • Symmetrische und asymmetrische Kryptoverfahren (DES, RSA, Betriebsmodi, One-Time-Pad, Hashfunktionen, Message-Authentication-Code, Elliptischen Kurven, Schlüsselerzeugung und -austausch) • Public-Key-Infrastruktur (öffentliche und geheime Schlüssel, Trust Center, Zertifikate und Zertifikathierarchien, PKI-Komponenten, Schlüsselmanagement) • Kryptographische Protokolle und Anwendungen (E-Commerce-Sicherheit, Copyright & Privacy Protection) • Sicherheit in Netzen (Paketfilter, Proxy-Server, Application-Gateway, sichere Kommunikation und sichere Anwendungsdienste) | | | | | |
| Literatur Patrick Horster: Kryptologie - BI-Reihe Informatik/47, 1988 Wolfgang Ertel: Angewandte Kryptographie, Fachbuchverlag, 2007 Bruce Schneier: Applied Cryptography, John Wiley & Sons, 1996 Claudia Eckert: IT-Sicherheit, Oldenbourg Verlag, 2008 | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter als PDF | | | | | |

| | | | | | |
|---|--|-------------|---|--|--|
| Modulbezeichnung / Prüfungsfach Selected Topics in Applied Computer Science | | Liste IB | Modulnummer 7250 | Sem. 4, 5 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben) | | | Dozenten Dozenten des Studienbereichs, Gastprofessoren, Lehrbeauftragte | | Verantwortlich Gergeleit |
| Inhaltliche Voraussetzungen | | | | | |
| Lernziele Explore current topics in applied computer science. • Use existing skills to acquire knowledge in new, related topics. • Implement application scenarios using modern concepts and technologies. | | | | | |
| Inhalte Contents of course depends on selected topic. | | | | | |
| Literatur Current literature relating to selected topic | | | | | |
| Medienform / Unterrichtssprache • Script, slides, and exercise sheets • Course language is English | | | | | |

| | | | | | |
|--|--|-------------|---|--|--|
| Modulbezeichnung / Prüfungsfach Sensornetze | | Liste IB | Modulnummer 7260 | Sem. 4, 5 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben) | | | Dozenten Gergeleit, Kröger, Werntges | | Verantwortlich Werntges |
| Inhaltliche Voraussetzungen • Modul "Rechnernetze und Telekommunikation" oder das Buch Computernetzwerke, Andrew S. Tanenbaum: Grundkenntnisse über die Vernetzung von Rechnern | | | | | |
| Lernziele Die Automatisierung durchdringt nach der Industrie in zunehmendem Maße auch die Gebäudetechnik und die private Heimumgebung bis hin zum Ubiquitous Computing. Dabei spielt die preisgünstige und flexible Vernetzung einfacher Sensoren und Aktoren überall eine zentrale Rolle. Nach erfolgreichem Abschluss dieses Moduls • kennen die Studierenden die Grundlagen sowohl der klassischen Feld- und Installationsbusse als auch der drahtlosen Sensornetze • sind die Studierenden in der Lage, für konkrete Anwendungen und Randbedingungen eine angemessene Netzwerklösung zu konzipieren. • können die Studierenden mit gegebenen Sensorknoten selbst größere Netze implementieren und prototypische Datenerfassungen, bzw. Steuerungsanwendungen realisieren. Die erworbenen Fähigkeiten erweitern damit die technologischen Kompetenzen im Bereich der technischen Informatik und der Automatisierungstechnik und vertiefen die Design- und Realisierungskompetenzen bzgl. der Entwicklung systemnaher Software. | | | | | |
| Inhalte • Grundlagen und Architektur der Sensornetze • Mechanismen und Protokolle (Bitübertragung, Sicherungsschicht, Vermittlungs- und Transportschicht) • Vergleich der Netzwerktechniken • Ad-hoc Netze • Arten von Sensoren • Architektur der Sensornetzknotten • Echtzeitanforderungen und Zeitsynchronisation • Anwendungen von Sensornetzen • Beispiele für Feldbusse (z.B. CAN, Profibus), Installationsbusse (z.B. EIB, LON, LCN) und drahtlose Sensornetze (z.B. ZigBee) | | | | | |
| Literatur Holger Karl, Andreas Willig: Protocols and Architectures of Wireless Sensor Networks, - Wiley, 2005 Schnell, Wiedemann (Herausgeber): Bussysteme in der Automatisierungs- und Prozesstechnik: Grundlagen, Systeme und Trends der industriellen Kommunikation, Vieweg+Teubner, 2008 | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter als PDF | | | | | |

| | | | | | |
|---|--|-------------|---------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Signalverarbeitung | | Liste IB | Modulnummer 7270 | Sem. 4, 5 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben) | | | Dozenten Geib, Kaiser, Reith | | Verantwortlich Geib |
| Inhaltliche Voraussetzungen • Grundlagen der Analysis, Integral- und Differenzialrechnung | | | | | |
| Lernziele Der Studierende • lernt die digitale Verarbeitung analoger Signale und deren Eigenschaften kennen und anwenden • kann die Zusammenhänge zwischen Zeitbereich- und Frequenzbereichdarstellungen (Integraltransformationen) deuten • kann die Auswirkungen bei der nicht idealen Abstastung/Messung einschätzen • kann Filter und Signalumsetzer (ADU, DAU) entwerfen und lernt Modulationen kennen • beherrscht Realisierungskonzepte der Signalverarbeitung mit modernen Rechensystemen und Signalprozessoren | | | | | |
| Inhalte • Eigenschaften der Signale (Energie und Leistung) • Lineare zeitinvariante Systeme (Linearität, Zeitinvarianz und Kausalität) • System- und Übertragungsfunktion, Fouriertransformation und ihre Eigenschaften • Bandbreite und Kanalkapazität • Abstastung (Zeit- und Frequenzbereich) und Rekonstruktion • Korrelation • z-Transformation • Filter (Tief-, Band- und Hochpass, signalangepasste Filter), AD/DA-Wandler und Multiplexer • Modulation | | | | | |
| Literatur Schneider-Obermann und Mildenerger, Basiswissen der Elektro-, Digital- und Informationstechnik: Für Informatiker, Elektrotechniker und Maschinenbauer, Vieweg+Teubner, 2006 von Grünigen, Digitale Signalverarbeitung, Hanser Fachbuch, 2008 Lüke, Signalübertragung, Springer, 1995 Streans, Digitale Verarbeitung analoger Signale, Oldenbourg, 2002 Unbehauen, Systemtheorie 1 und 2, Oldenbourg, 2002 | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript, Folien und Übungsblätter | | | | | |

| | | | | | |
|---|--|-------------|--------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Skriptsprachen | | Liste IB | Modulnummer 7280 | Sem. 4, 5 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 60 h Bearbeitung (Praktikumsaufgaben) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Anwesenheit (Vorlesung und Praktikum) | | | Dozenten Barth, Werntges | | Verantwortlich Werntges |
| Inhaltliche Voraussetzungen | | | | | |
| Lernziele Höhere, mächtigere Sprachparadigmen und auf den Aufgabenbereich zugeschnittene Sprachen erlauben es, Anwendungen effizienter und kompakter zu realisieren und vorhandene Lösungen und Dienste einfach zu integrieren. Absolventen des Moduls können • höhere Sprachparadigmen und mächtige Datenstrukturen sicher einsetzen • praxisrelevante Aufgaben unter Nutzung fertiger Bibliotheken schnell umsetzen oder prototypisch realisieren • verschiedene praxisrelevante Vertreter von Skriptsprachen dem Einsatzzweck gemäß auswählen und effizient einsetzen Die erworbenen Fähigkeiten tragen damit in besonderem Maße zur Ausprägung von spezifischen Analyse-, Design- und Realisierungskompetenzen und zur Erweiterung von spezifischen technologischen Kompetenzen bei. | | | | | |
| Inhalte Grundlagen: • Historie, Überblick, Eigenschaften von Skriptsprachen • Kombination vorhandener Werkzeuge am Beispiel Shell Skript • Kommandozeile, wichtige Unix-Tools und Shell-Skript Tools • Variablen, Kontrollstrukturen, Funktionen, Parameter, Stringverarbeitung • Typische Beispiele • Reguläre Ausdrücke • Mächtige, universelle, einfache Skriptsprache an einem aktuellen Beispiel • Einbettung mächtiger Datenstrukturen (Listen, Dictionaries, Tupel) • Mächtige Kontrollstrukturen, schlanke Syntax, dynamische Typisierung • Objektorientierung, Ausnahmen und Modulkonzept • Funktionales Programmieren • Typische Einsatzgebiete (Prototyping, Testing) an konkreten Beispielen • Nutzen von integrierten Funktionalitäten und hoch abstrahierten Bibliotheken (Stringverarbeitung, reguläre Ausdrücke, GUI-Programmierung mit zum Beispiel Tkinter, wxPython, FXRuby, etc.) Ausgewählte Skriptsprachen-Aspekte wie: • Web-Programmierung am Beispiel PHP • Web-Infrastruktur, Server-side und Client-side Scripting, • Client-Side Scripting am Beispiel JavaScript • Einfache Persistenz-Techniken, ORM-Anbindung an Datenbanken • Extensions zur Integration von Java- bzw. C/C++-Bibliotheken • Domain specific languages • Anwendungsintegration | | | | | |
| Literatur Lutz, Ascher: Einführung in Python, O'Reilly Cooper: Advanced Bash Scripting Guide, http://www.tldp.org/LDP/abs/html/ Gunnar Thies, Stefan Reimers: PHP 5.3 und MySQL 5.1: Grundlagen, Anwendung, Praxiswissen, Objektorientierung, MVC, Sichere Webanwendungen, PHP-Frameworks, Performancesteigerungen, Galileo Press; 2009 D. Thomas et al.: Programming Ruby 1.9. The Pragmatic Programmers' Guide, The Pragmatic Bookshelf; 2009 M. Odersky et al.: Programming in Scala, Artima Press; 2008 | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter als PDF | | | | | |

| | | | | | |
|--|--|-------------|--|--|--|
| Modulbezeichnung / Prüfungsfach Usability Engineering | | Liste IB | Modulnummer 7290 | Sem. 4, 5 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben) | | | Dozenten Dörner, Schulz, Schwanecke | | Verantwortlich Dörner |
| Inhaltliche Voraussetzungen • Objektorientierte Softwareentwicklung: Programmierkenntnisse • Computergrafik: Grundkenntnisse der generativen Computergrafik und über GDV-Systeme • Softwaretechnik: Grundkenntnisse von Methodiken der Softwaretechnik (wie Vorgehensmodelle, Modellierung, Use Cases) | | | | | |
| Lernziele Die Studierenden sollen • Usability als ein wesentliches Kriterium für die Qualität von Software begreifen. • Fähigkeiten entwickeln, um Usability bei graphisch-interaktiven Systemen zu erzielen und bewerten zu können. • Methodiken kennen lernen, wie die Usability von Software im Softwareentwicklungsprozess gezielt erreicht und überprüft werden kann. | | | | | |
| Inhalte • Benutzbarkeit von Software • Zielsetzungen der Software Ergonomie • Grundlagen von Wahrnehmung und Gedächtnis • Handlungsprozesse beim Umgang mit Software • Dialogformen und Interaktionsdesign • Grafische Dialogsysteme, Usability bei Grafisch-Interaktiven Systemen • Normen, Gesetze, Richtlinien • Usability Testing • Usability Engineering im Softwareentwicklungsprozess • User-Centered Design • Prototyping | | | | | |
| Literatur M. Dahm: Grundlagen der Mensch-Computer-Interaktion, Pearson Studium, 2006 D. Mayhew: The Usability Engineering Lifecycle, Morgan Kaufmann, 1999 B. Shneiderman, C. Plaisant, M. Cohen: Designing the User Interface, Pearson Education, 2009 | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter (als pdf-Dateien) | | | | | |

| | | | | | |
|---|--|-------------|--|--|--|
| Modulbezeichnung / Prüfungsfach Virtual Reality-Systeme | | Liste IB | Modulnummer 7300 | Sem. 4, 5 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben) | | | Dozenten Dörner, Schulz, Schwanecke | | Verantwortlich Dörner |
| Inhaltliche Voraussetzungen • Objektorientierte Softwareentwicklung: Programmierkenntnisse • Computergrafik: Grundkenntnisse der generativen Computergrafik und über GDV-Systeme | | | | | |
| Lernziele Virtuelle Realität ist eine wesentliche Anwendung der Echtzeit-Computergrafik und hat in einer ganzen Bandbreite verschiedener Erscheinungsformen (von Desktop VR bis Full Immersive VR) zahlreiche Anwendungen. Die Studenten sollen das nötige Rüstzeug für die Erstellung von VR kennen und einsetzen können – sowohl algorithmisch als auch technologisch. Daneben sollen Studenten auch die Grundlagen für die erfolgreiche Illusion von Immersion kennen und die Einsatzmöglichkeiten von VR in einzelnen Anwendungen als auch für die Gestaltung von User Interfaces beurteilen können. | | | | | |
| Inhalte • Immersion und Präsenz, Raumwahrnehmung • Spezielle Ein- und Ausgabegeräte • Tracking • Aktives und passives Stereo • Features von VR-Systemen (z.B.Kollisionsdetektion) • Sprach- und Gesteninteraktion • Integration von 3D Audio, Audioeffekte für VR (z.B. Echo, Hall) • VR Installationen, VR und Augmented Reality (AR) • Architektur von VR und AR Systemen • Anwendungen (Virtual prototyping, Walk Through) | | | | | |
| Literatur G. Kim: Designing Virtual Reality Systems: The Structured Approach, Springer, 2005 | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter (als pdf-Dateien) | | | | | |

| | | | | | |
|--|--|-------------|---|--|--|
| Modulbezeichnung / Prüfungsfach Wirtschaftsinformatik | | Liste IB | Modulnummer 7310 | Sem. 4, 5 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 60 h Bearbeitung (Praktikumsaufgaben) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Anwesenheit (Vorlesung und Praktikum) | | | Dozenten Heimrich, Hünemohr, Krechel | | Verantwortlich Krechel |
| Inhaltliche Voraussetzungen • Datenbanken: Modellierung, Transaktionskonzept • Betriebswirtschaftslehre: Betriebswirtschaftliche Grundbegriffe und Grundtatbestände | | | | | |
| Lernziele Die Wirtschaftsinformatik befasst sich mit der Nutzung der Informations- und Kommunikationstechnik in Wirtschaft und Verwaltung, insbesondere mit der Entwicklung und dem Einsatz betrieblicher Anwendungs- und Informationssysteme. Ziel der Lehrveranstaltung ist es, die Konzepte und Architektur betriebswirtschaftlicher Informationssysteme - insbesondere am Beispiel eines integrierten ERP-Systems - zu untersuchen. Auch werden praxisorientiert die Methoden und Vorgehensweisen zur Analyse, Modellierung und Optimierung von Geschäftsprozessen - am Beispiel eine Business Process Modeling/Management (BPM) Tools - behandelt. Die Studierenden können nach Besuch der Lehrveranstaltung: • Geschäftsprozesse analysieren und modellieren (Praxisbeispiel BPM-Tools) • die Architektur und den Einsatz betrieblicher Informationssysteme bewerten (Praxisbeispiel ERP-System) • betriebliche Kernprozesse in ERP-Systemen zuordnen und strukturieren • die Bedeutung des IT-gestützten Informationsmanagements eines Unternehmens und die IT-Sicherheit im E-Business bewerten | | | | | |
| Inhalte Die Lehrveranstaltung behandelt ausgewählte Themen der folgende Gebiete der Wirtschaftsinformatik: • Wirtschaftsinformatik - Anwendungsgebiet der Informatik • Aktuelle Themengebiete der Wirtschaftsinformatik • Theorie und Praxis der Geschäftsprozessmodellierung anhand von Anwendungsbeispielen aus der Wirtschaft • Praxisbeispiel eines Werkzeugs für die Geschäftsprozessanalyse (BPA),-modellierung und Geschäftsprozessmanagement (BPM) • Anwendungsgebiete des Business Process Reengineering • Architektur und Einsatz von integrierten betrieblichen Informationssystemen • Unterstützung der betrieblichen Kernprozesse mit ERP-Systemen • Praxisbeispiel eines ERP-Systems • Grundlage der Supply Chain und Customer Relationship Management-Systeme • Beispiele für E-Business/E-Commerce-Technologien und Anwendungssysteme • Rechtssichere Durchführung von elektronischen Kommunikations- und Transaktionsprozessen - die elektronische Signatur (Technik, Lösungen) | | | | | |
| Literatur Laudon: Wirtschaftsinformatik – Eine Einführung, Pearson, 2009 Scheer: ARIS – Vom Geschäftsprozess zum Anwendungssystem, Springer, 2002 Aktuelle Beiträge aus Fachzeitschriften (u.a. "Informatik Spektrum") | | | | | |
| Medienform / Unterrichtssprache • Web-Seite zur Veranstaltung • Skript/Folien Übungsblätter als PDF im Netz | | | | | |

| | | | | | |
|--|--|-------------|--------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach XML-Technologien und W3C-Standards | | Liste IB | Modulnummer 7320 | Sem. 4, 5 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 60 h Bearbeitung (Praktikumsaufgaben) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Anwesenheit (Vorlesung und Praktikum) | | | Dozenten Panitz, Werntges | | Verantwortlich Werntges |
| Inhaltliche Voraussetzungen • Unicode.org: Unicode, UTF-8, UTF-16 • WBA: HTML- und CGI-Grundkenntnisse | | | | | |
| Lernziele XML-basierte Auszeichnungssprachen insbesondere des W3C spielen eine immer größere Rolle bei der Repräsentation, dem maschinellen Austausch und der Weiterverarbeitung strukturierter Informationen, beispielsweise im elektronischen Datenaustausch, vielen Internet-Anwendungen und mobilen Informationssystemen. Nach bestandem Kurs besitzen die Teilnehmer die Fähigkeit • zur Nutzung von XML zur Definition von Auszeichnungssprachen, • sowie zur Erzeugung und Weiterverarbeitung XML-basierter Dokumente (insb. Parsing, Transformation) Sie können • die verschiedenen Rollen Autor, Dokumententyp-Designer sowie Gestalter klar unterscheiden und mit XML-Standards ausfüllen und damit die klare Trennung zwischen Inhalten und Darstellung konsequent umsetzen • XML in unterschiedlichen Einsatzbereichen (z.B. Multimedia, el. Datenaustausch) sinnvoll einsetzen Die erworbenen Fähigkeiten tragen in besonderem Maße zur Ausprägung von spezifischen Analyse-, Design- und Realisierungskompetenzen, zur Erweiterung von spezifischen technologischen Kompetenzen und zu fachübergreifenden technischen Systemkenntnissen bei. | | | | | |
| Inhalte • Einführung: Strukturierte Dokumente, Grundkonzepte von Auszeichnungssprachen, Anwendungsbeispiele, historische Entwicklung • Trennung der Aspekte Inhalt / Struktur / Darstellung eines Dokuments; Cascading Stylesheets (CSS) • HTML als Beispiel für eine konkrete Auszeichnungssprache; Grundbegriffe (Elemente, Attribute, Entities, Dokumententypdeklaration) • Metasprachen zur Definition von Auszeichnungssprachen am Beispiel von XML • XML Dokumententypdefinition (DTD), Unterscheidung "well-formed" / "valid" • Das XML-Namensraumkonzept (namespaces) • XML Infoset: Die abstrakte Sicht auf XML-Dokumente • Einführung in XML Schema: Überblick, Vergleich mit DTDs • Einsatz von XPath zur Identifikation von Dokumententeilen: Konzepte (Achsen, Pfadausdrücke, Funktionen) • Verarbeitung von XML-Dokumenten mit XSL: Verarbeitungsmodell von XSLT, XSLT-Stylesheets, Verwendung von XPath, XSLT-Sprachmittel einschließlich Kontrollstrukturen, Rekursion, parametrisierte Templates, Sortierung/Gruppierung; • Verknüpfung von XML-Dokumenten, Markierung und Klassifizierung mit XPointer und XLink; XBase • XML-Anwendungsbeispiele aus verschiedenen Bereichen (z.B. Vektorgraphik mittels SVG, Druckseitengestaltung mit XSL-FO) | | | | | |
| Literatur Goldfarb, Prescott: The XML Handbook, Prentice-Hall Hunter et al: Beginning XML, Wrox Kay: XSLT, Wrox Brett McLaughlin: Java & XML, O'Reilly | | | | | |
| Medienform / Unterrichtssprache • Web-Seite zur Veranstaltung • Folien / Übungsblätter als PDF | | | | | |

| | | | | | |
|---|--|-------------|--------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Unternehmensplanspiel | | Liste IB | Modulnummer 7330 | Sem. 4, 5 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 5 cp, 150 h 4 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben) | | | Dozenten Jarass | | Verantwortlich Jarass |
| Inhaltliche Voraussetzungen • LV 1311: Einführung in die Betriebswirtschaftslehre | | | | | |
| Lernziele Viele Informatikingenieure benötigen Grundkenntnisse der Wirtschaftswissenschaften bei ihrer Arbeit: • Strukturierung betrieblicher Entscheidungsprozesse • Planung und Realisierung von Produktion und Absatz • Planung der Finanzierung und Sicherstellung der Liquidität • Kostenzuordnung und Kostenoptimierung | | | | | |
| Inhalte Entsprechende Vorkenntnisse in BWL werden im Rahmen eines computergestützten Unternehmensplanspiels vertieft: • Jeweils 3 Studenten bilden ein Unternehmen (Produktion, Beschaffung, Absatz, Finanzierung, Personal), die mit einem Produkt auf einem Markt mit anderen Unternehmen konkurrieren. • Das verwendete Planspiel CABA2000 wurde an der Hochschule RheinMain speziell für Ingenieurstudenten entwickelt. • Das Planspiel erfordert eine außerordentlich intensive Mitarbeit der Studenten. | | | | | |
| Literatur Jarass, L.: CABA2000 - Spielerhandbuch und Übungshandbuch (abrufbar unter www.JARASS.com , Lehre). | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite (www.JARASS.com) • Die LV wird als internetfähiges Planspiel durchgeführt | | | | | |

| | | | | | |
|---|--|-------------|--|--|--|
| Modulbezeichnung / Prüfungsfach 2D-Bildanalyse | | Liste SB | Modulnummer 7500 | Sem. 4 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 10 cp, 300 h 6 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (4 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt) | | | Dozenten Dörner, Schulz, Schwanecke | | Verantwortlich Schwanecke |
| Inhaltliche Voraussetzungen • Analysis: Differential- und Integralrechnung mehrdimensionaler Funktionen, partielle Ableitungen, Gradienten • Lineare Algebra: Lineare Gleichungssysteme, Matrizen, Eigenwerte, Eigenvektoren • Objektorientierte Softwareentwicklung: Programmierkenntnisse | | | | | |
| Lernziele Die 2D Bildanalyse (oft auch Bildverstehen genannt) beschäftigt sich mit der Analyse und Interpretation von Bildern bzw. Bildfolgen. Die Hauptaufgabe der Bildanalyse besteht darin, auf den Bildern etwas (zum Beispiel Objekte, die sich vom Hintergrund abheben) zu erkennen. Die Studierenden sollen die wichtigsten Verfahren im Bereich der 2D Bildanalyse kennen lernen und einsetzen können. | | | | | |
| Inhalte • Bildvorverarbeitung • Objektmerkmale • Klassifikationsprinzipien • Numerische Klassifikation (lineare Klassifikation, Abstandsklassifikation) • Statistische Klassifikation • Syntaktische Klassifikation • Kontextabhängige Klassifikation (Graphmatching, diskrete und kontinuierliche Relaxation) • Hauptkomponentenanalyse • Bewegungsdetektion und Objektverfolgung • Bayessche Filter • Anwendungen (Gesichtserkennung, Inhaltsbasierte Bildsuche) | | | | | |
| Literatur Steinmüller: Bildanalyse, Springer, 2008 Tönnies: Grundlagen der Bildverarbeitung, Pearson, 2005 Gonzales, Woods: Digital Image Processing, Prentice Hall, 2008 | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter (als pdf-Dateien) | | | | | |

| | | | | | |
|---|--|-------------|------------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Advanced Networking | | Liste SB | Modulnummer 7510 | Sem. 4 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 10 cp, 300 h 6 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (4 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt) | | | Dozenten Geib, Gergeleit, Weitz | | Verantwortlich Gergeleit |
| Inhaltliche Voraussetzungen • Modul "Rechnernetze und Telekommunikation" oder das Buch Computernetzwerke, Andrew S. Tanenbaum: Kenntnisse über die Vernetzung von Rechnern, speziell IP-Netze | | | | | |
| Lernziele Die Protokolle und Dienste der Sprach- und Datennetze sind heute eine Basistechnologie für den Aufbau moderner Kommunikationssysteme und damit der gesamten Informationsgesellschaft. Daher werden in diesem Modul diese Konzepte der Übertragungs- und Vermittlungstechnik vertieft sowie beispielhaft anhand von konkreten Systemen vorgestellt. Dieses Modul befähigt die Studierenden • den strukturierten Aufbau von integrierten Sprach- und Datennetzen und die Funktionen, Dienste und Protokolle von aktuellen Kommunikationsnetzen zu kennen, • die Leistung des Gesamtsystems und die Zusammenarbeit der Komponenten in solchen Netzen verstehen und beurteilen zu können, • in der Lage zu sein, selbst komplexere Netzwerkinfrastrukturen konzipieren und konfigurieren zu können, • und diese Kenntnisse in einem umfangreicheren Semester-Projekt zu Lösung eines konkreten Problems praktisch anwenden zu können. Neben dem Erwerb dieser spezifischen Analyse-, Design- und Realisierungskompetenzen sowie der technischen Kompetenzen haben sie ihre kommunikative und soziale Kompetenzen durch Präsentation von eigenen Projektergebnissen und durch die Arbeit in kleinen Projektteams vertieft. | | | | | |
| Inhalte • Lokale Netze (IEEE 802 LANs) • Drahtlose Lokale Netze (IEEE 802.11 WLANs) • Routing und Routingprotokolle (RIP, OSPF, BGP) • Switching (ATM, MPLS) • "Last-Mile" Zugangstechnologien (XDSL, Funktechnologien) • Zellulare Mobilkommunikationsnetze (GSM/GPRS, UMTS) • Integrierte Daten- und Sprachnetze (NGNs, VoIP) • Dienstgüearchitekturen (IntServ, DiffServ) • Authentifikation, Vertraulichkeit, Nachrichtenintegrität auf verschiedenen Ebenen (IEEE 802.1x, EAP, IPsec, TLS) • Netzwerk-Monitoring und -Management • Durchführung praktischer Experimente mit Netzwerk-Infrastrukturkomponenten | | | | | |
| Literatur Kurose, Ross: Computernetzwerke: Der Top-Down-Ansatz, 4. Auflage, Pearson Studium, 2008 Siegmond: Technik der Netze: Neue Ansätze: SIP in IMS und NGN, Hüthig, 2009 | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript, Vorlesungsfolien und Übungsblätter | | | | | |

| | | | | | |
|---|--|-------------|------------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Anwendungen der Künstlichen Intelligenz | | Liste SB | Modulnummer 7520 | Sem. 4 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 10 cp, 300 h 6 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (4 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt) | | | Dozenten Barth, Krechel, Panitz | | Verantwortlich Krechel |
| Inhaltliche Voraussetzungen | | | | | |
| Lernziele Die Studenten sind in der Lage, geeignete Suchstrategien zur Lösungsermittlung auszuwählen und in Teams erfolgreich eine intelligente Applikation zu entwickeln. Mobile Geräte mit verschiedenen Sensoren wie Mobiltelefone und Mobile autonome Roboter werden erfolgreich in die Applikationen über Serviceschnittstellen integriert. | | | | | |
| Inhalte <ul style="list-style-type: none"> • Grundlagen der Modernen KI • Ansatz des intelligenten Agenten • Verschiedene Suchstrategien theoretisch und beispielhaft • Implementierung ausgesuchter Suchstrategien in passenden Programmiersprachen • Aktuelle Forschungs- und Anwendungsprojekte in Teamarbeit • eventbasierte Serviceschnittstellen • Middleware • Elemente der Robotik • autonome Service Roboter • Sensoren (Video, Audio) • Elemente der Bildverarbeitung, Sprachverarbeitung und Mustererkennung | | | | | |
| Literatur Karsten Berns: "Autonomous Land Vehicles", Vieweg+Teubner 2009 Russel Norvig: "Künstliche Intelligenz", Pearson 2004 | | | | | |
| Medienform / Unterrichtssprache <ul style="list-style-type: none"> • veranstaltungsspezifische Webseite • Skripte in elektronischer Form | | | | | |

| | | | | | |
|--|--|-------------|--|--|--|
| Modulbezeichnung / Prüfungsfach Computer Vision | | Liste SB | Modulnummer 7530 | Sem. 4 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 10 cp, 300 h 6 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (4 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt) | | | Dozenten Dörner, Schulz, Schwanecke | | Verantwortlich Schwanecke |
| Inhaltliche Voraussetzungen • Analysis: Differential- und Integralrechnung mehrdimensionaler Funktionen, partielle Ableitungen, Gradienten • Lineare Algebra: Lineare Gleichungssysteme, Matrizen, Eigenwerte, Eigenvektoren • Objektorientierte Softwareentwicklung: Programmierkenntnisse | | | | | |
| Lernziele Die Computer Vision (oder auch das Maschinelle Sehen) beschäftigt sich mit der rechnergestützten Lösung von Aufgabenstellungen, welche üblicherweise mit Hilfe des menschlichen visuellen Systems bearbeitet werden. Maschinell sehende Systeme haben einen immer größeren Einfluss auf unser tägliches Leben und finden sich zum Beispiel neben der Automatisierungstechnik und Qualitätskontrolle inzwischen auch in Bereichen wie der Verkehrs- oder Sicherheits-Technik. Die Studierenden sollen das nötige Grundlagenwissen für das Verständnis von Algorithmen und Verfahren aus dem Bereich der Computer Vision erwerben und einsetzen können. | | | | | |
| Inhalte • Klassifikation unterschiedlicher Geometrien in 1D, 2D und 3D • Registrierung von 2D und 3D Daten • Kegelschnitte und Quadriken • Winkelmessung im projektiven Raum • Bildrektifizierung • Kamera-Modelle • Kamera-Kalibrierung • Stereogeometrie • Matrizenzerlegungen (QR, Cholesky, Singulärwertzerlegung) • Kleinste Quadrate Lösungen linearer Gleichungssysteme • RANSAC • N-Ansichten Geometrie • Shape from X | | | | | |
| Literatur Hartley, Zissermann: Multiple View Geometry in Computer Vision, Cambridge, 2004 Forsyth, Ponce: Computer Vision A Modern Approach, Prentice Hall, 2003 Bradski, Kaehler: Learning OpenCV: Computer Vision with the OpenCV Library, O' Reilly, 2008 | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter (als pdf-Dateien) | | | | | |

| | | | | | |
|---|--|-------------|--------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Content- und Wissensmanagement | | Liste SB | Modulnummer 7540 | Sem. 4 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 10 cp, 300 h 6 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (4 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 150 h Praktische Tätigkeit (Projekt) • 60 h Vor- und Nachbereitung (Theorieteil) • 90 h Anwesenheit (Vorlesung und Praktikum) | | | Dozenten Barth, Krechel | | Verantwortlich Krechel |
| Inhaltliche Voraussetzungen • Datenbanken: Relationale Datenbanken | | | | | |
| Lernziele Inhalte • Einführung in die Konzepte des prozessorientierten Wissensmanagement - Dokumentenmanagement • BPM und Workflowmanagement • Web-Contentmanagement • Suche in Informationsbeständen • Klassifikation und Extraktion | | | | | |
| Inhalte Content-Management-Systeme (CMS) werden vermehrt zur Ablage und zum effizienten Retrieval immer häufiger elektronisch verwalteter unstrukturierter Daten eingesetzt und mit strukturierten Daten sinnvoll verknüpft. • Standards und Techniken zur Verwaltung unstrukturierter und strukturierter Daten gezielt einsetzen • Einsatzbereich von CMS bestimmen, CMS auswählen und praktisch verwenden • Techniken zur Wissensextraktion, -Darstellung und Verwaltung kennen und einsetzen | | | | | |
| Literatur Abecker, et al.: Geschäftsprozessorientiertes Wissensmanagement: Effektive Wissensnutzung bei der Planung und Umsetzung von Geschäftsprozessen, Springer, 2002 Lehner: Wissensmanagement. Grundlagen, Methoden und technische Unterstützung, Hanser, 2009 Götzer, Schmale, Maier, Komke: Dokumenten-Management: Informationen im Unternehmen effizient nutzen, dpunkt, 2004 Gadatsch: Grundkurs Geschäftsprozess-Management: Methoden und Werkzeuge für die IT-Praxis, Vieweg, 2007 Koop: Erfolgsfaktor Content Management. Vom Web Content bis zum Knowledge Management, Vieweg, 2001 | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript, Folien und Übungsblätter | | | | | |

| | | | | | |
|---|--|-------------|--------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Data Warehouse-Systeme und Data Mining | | Liste SB | Modulnummer 7550 | Sem. 4 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 10 cp, 300 h 6 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (4 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt) | | | Dozenten | | Verantwortlich Muth |
| Inhaltliche Voraussetzungen • Datenbanken | | | | | |
| Lernziele Die Studierenden sollen in der Lage sein: • ein multidimensionales datenmodell zu entwerfen • ein Data-Warehouse-System einzurichten, zu warten und zu optimieren Die Studierenden sollen die Grundlagen von Data Mining Verfahren kennen. Sie sollen in der Lage sein, passende Data Mining Verfahren auszuwählen und diese korrekt anzuwenden. | | | | | |
| Inhalte • Referenzarchitektur von Data Warehouse Systemen • Phasen des Data Warehousing • Physische Architektur • Das multidimensionale Datenmodell und dessen Umsetzung • Optimierung des Datenmodells und der Anfragen • Vorgehensweise beim Aufbau eines Data Warehouse Systems • Betrieb und Weiterentwicklung eines Data Warehouse Systems • Data Mining Grundlagen (Statistik) • Data Mining Verfahren | | | | | |
| Literatur Bauer, Günzel: Data Warehouse Systeme - Architektur, Entwicklung, Anwendung, dpunkt.verlag Lehner: Datebanktechnologie für Data-Warehouse-Systeme, dpunkt.verlag Schrödl: Business Intelligence mit MS SQL Server 2008, Hanser-Verlag | | | | | |
| Medienform / Unterrichtssprache • Web-Seite zur Veransattung • Folien/Skript • Projektaufgabe | | | | | |

| | | | | | |
|--|--|-------------|--------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach E-Business: Standards und Automatisierung | | Liste SB | Modulnummer 7560 | Sem. 4 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 10 cp, 300 h 6 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (4 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt) | | | Dozenten Werntges | | Verantwortlich Werntges |
| Inhaltliche Voraussetzungen | | | | | |
| <p>Lernziele</p> <p>Heutige Unternehmen nutzen nur einen Bruchteil der E-Business-Möglichkeiten. Eine wichtige Ursache: Mangelnde Schlüsselqualifikationen der beteiligten Fachkräfte. Dieser Kurs strebt an, einige dieser Lücken gezielt zu schließen.</p> <p>Absolventen des Kurses besitzen</p> <ul style="list-style-type: none"> • die Fähigkeit, die heutige Vielfalt technischer wie fachlicher E-Business-Standards zu überblicken, ihre Abhängigkeiten voneinander zu erkennen und ihre Bedeutung einzuschätzen. • die Kenntnis organisatorischer Abhängigkeiten und Voraussetzungen für erfolgreiche und standardkonforme technische Umsetzungen. • die Fähigkeit, unternehmensübergreifende Geschäftsprozesse nach internationalen Standards technisch wie organisatorisch zu implementieren (Schwerpunkt). <p>Sie vertieften ihre Fertigkeiten in Projektmanagement- und sozialer sowie Selbst-Kompetenz durch das integrierte Projekt mit verteilten Rollen.</p> <p>Die erworbenen Fähigkeiten tragen ferner in besonderem Maße zur Erweiterung von spezifischen technologischen Kompetenzen und zu fachübergreifenden technischen Systemkenntnissen bei.</p> | | | | | |
| <p>Inhalte</p> <ul style="list-style-type: none"> • Technische Grundlagen: XML DTDs, Namensräume, XML Schema • Geschäftsprozess-Szenario, Übersicht fachliche und technische Standards • Organisatorische Voraussetzungen und Hindernisse • Ident-Systeme und Auto-ID Verfahren • Klassifikationsstandards, Katalogdaten-Standards • Transaktionsstandards, insbesondere UN/EDIFACT • ebXML und RosettaNet - Automatisierung ganzer Geschäftsprozesse • Reliable Messaging, techn. Grundlagen, Mailboxing, Messaging, File Transfer • Mapping • Umfeld im praktischen Einsatz: Monitoring und Alarmierung, Tracking und Tracing • Simulation logistischer Prozesse im Rahmen eines Planspiels | | | | | |
| <p>Literatur</p> <p>E-Business Standards in Deutschland: Bestandsaufnahme, Probleme, Perspektiven, Berlecon Research, 2003 (http://www.berlecon.de)</p> <p>EANCOM-CDROM (wird gestellt, mit freundlicher Unterstützung der GS1 Germany GmbH)</p> <p>G. Schmied: High Quality Messaging and Electronic Commerce. Springer, 1999.</p> | | | | | |
| <p>Medienform / Unterrichtssprache</p> <ul style="list-style-type: none"> • Web-Seite zur Veranstaltung • Folien, Übungsblätter • Stud.IP-Angebot der Hochschule | | | | | |

| | | | | | |
|--|--|-------------|--|--|--|
| Modulbezeichnung / Prüfungsfach Echtzeit-Computergraphik | | Liste SB | Modulnummer 7570 | Sem. 4 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 10 cp, 300 h 6 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (4 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt) | | | Dozenten Dörner, Schulz, Schwanecke | | Verantwortlich Dörner |
| Inhaltliche Voraussetzungen • Objektorientierte Softwareentwicklung: Programmierkenntnisse • Computergrafik: Grundkenntnisse der generativen Computergrafik und über GDV-Systeme | | | | | |
| Lernziele Die Studierenden sollen • grundlegende Kenntnisse über komplexe, hoch-interaktive, echtzeitfähige und Grafik-lastige Softwaresysteme (wie z.B. Computerspiele) erwerben • komplexe Echtzeit-Grafiksysteme entwerfen und realisieren können • dabei auch entsprechende Programmbibliotheken einsetzen und softwaretechnisch in das System integrieren können • die wichtigsten Datenstrukturen und Algorithmen zur Grafikbeschleunigung kennen, und in ihrer Wirkung beurteilen und in einer geeigneten Kombination einsetzen können | | | | | |
| Inhalte • Grafikkibliotheken und APIs für Echtzeitanwendungen • Echtzeitgrafiksysteme als Softwaresysteme - Eigenschaften und Erstellungsprozesse • Game Engines und Softwaretechnik • Beispiele von Echtzeitgrafiksystemen • Echtzeitfähigkeit von Renderingansätzen • Verhaltenssimulation in Echtzeit • Anwendungen von Echtzeitgrafiksystemen | | | | | |
| Literatur T. Akenine-Möller, E. Haines, N. Hoffmann: Real-Time Rendering, Peters Welleley, 2008 D. H. Eberly: 3D Game Engine Design. A Practical Approach to Real-Time Computer Graphics, Morgan Kaufmann, 2006 Ausgewählte Originalliteratur | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter (als pdf-Dateien) | | | | | |

| | | | | | |
|--|--|-------------|--------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Embedded Systems | | Liste SB | Modulnummer 7580 | Sem. 4 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 10 cp, 300 h 6 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (4 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt) | | | Dozenten Kaiser, Kröger | | Verantwortlich Kröger |
| Inhaltliche Voraussetzungen • Modul Grundlagen der Techn. Informatik: Rechnerorganisation • Modul Programmiermethodik und Programmierertechniken: Programmieren in C, Java • Modul Rechnernetze und Telekommunikation: Grundlagen Rechnernetze • Modul Betriebssysteme und Verteilte Systeme: Grundlagen Betriebssysteme | | | | | |
| Lernziele Lösungen der technischen Informatik durchdringen zunehmend die Produkte nahezu aller Branchen und sind gefragt, um die technische Wettbewerbsfähigkeit auf dem Weltmarkt zu sichern. Ein großer Teil der Funktionalität Eingebetteter Systeme wird mit Hilfe von Software erbracht. Die Entwicklung dieser Software unterscheidet sich aber wesentlich von klassischer Desktop/Server-Software, u.a. wegen der geforderten Verlässlichkeit, der nur begrenzt vorhandenen Ressourcen sowie der Variationsbreite der Hardware-Plattformen und der Lösungen. Nach erfolgreichem Abschluss des Moduls • sind die Studierenden in der Lage, charakteristische Unterschiede zwischen Desktop-/Server-basierter Software und der Software Eingebetteter Systeme zu erkennen und Konsequenzen für die Softwareentwicklungsmethoden abzuleiten. • können sie Ausführungsplattformen hinsichtlich ihrer Eignung für die Gesamtfunktionalität bewerten und auswählen. • sind die Studierenden fähig, hardwarenahe Software zu entwickeln. • können sie quantitative Bewertungen der verfügbaren Ressourcen (Speicher, Energie, Rechenleistung, eingeschränkte Netzwerkkonnektivität) vornehmen und mit der geforderten Softwarefunktionalität abgleichen. • beherrschen die Studierenden Techniken zur Realisierung verlässlicher Software und von vernetzten Eingebetteten Systemen. • können sie Projektergebnisse präsentieren • können sie Zusammenarbeit in kleinen Projektteams organisieren Insgesamt trägt das Modul zu einer Erweiterung der technologischen Kompetenzen im Bereich der Informatik der Systeme bei, vertieft die Design- und Realisierungskompetenzen und fördert die kommunikativen Kompetenzen sowie die Projektmanagement-Kompetenz. | | | | | |
| Inhalte • Hardwarenahe Programmierung von Eingebetteten Systemen • Metriken zur quantitativen Bewertung von Anforderungen und von Ausführungsplattformen sowie deren Abgleich • Konfigurierbarkeit von Software, aspektorientierte Programmierung, Software-Produktlinien • Betriebssysteme für Eingebettete Systeme • Software-Vernetzung von Eingebetteten Systemen, Kommunikations-Middleware • Fehlererkennung, Ausnahmebehandlung, Robustheit, Fehlertoleranz • Zuverlässige Software: Standards und Normen, Zertifizierung • Modellbasierter Entwurf von Software für Eingebettete Systeme, Software-Synthese • Test- und Verifikationstechniken für Software Eingebetteter Systeme Es werden spezielle Themen aus diesen Bereichen an kleine Gruppen von 2-3 Studierenden vergeben. | | | | | |
| Literatur P. Marwedel: Eingebettete Systeme, Springer, 2008 H. Wörn, U. Brinkschulte: Echtzeitsysteme, Springer, 2005 H. Kopetz: Real-Time Systems - Design Principles for Distributed Embedded Applications, Kluwer Acad. Publ., 1997 Weitere Literatur wird jeweils spezifisch bekannt gegeben. | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Projektaufgabe in schriftlicher Form | | | | | |

| | | | | | |
|---|--|-------------|---------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Fortgeschrittene Softwaretechnologie | | Liste SB | Modulnummer 7590 | Sem. 4 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 10 cp, 300 h 6 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (4 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt) | | | Dozenten Barth, Iglér, Weitz | | Verantwortlich Iglér |
| Inhaltliche Voraussetzungen • Softwaretechnik: Analyse, Design-Patterns • Gute Programmierkenntnisse (OO) in Java • Datenbanksysteme: Gute Kenntnisse über relationale Datenbanken • Webanwendungen: Web-Komponenten der Java Enterprise Plattform | | | | | |
| Lernziele Entwicklung mehrschichtiger "Enterprise"-Web-Anwendungen unter Einsatz von Software-Komponenten. • Analyse einer mehrschichtigen Web-basierten Anwendung • Design unter Einsatz dafür geeigneter Design-Patterns und Entwicklungs-Frameworks • Implementierung und Deployment auf einem geeigneten Anwendungs-Server • Einsatz eines geeigneten CASE-Tools und/oder Entwicklungswerkzeuges | | | | | |
| Inhalte • Überblick Komponententechnologien • Komponenten-Frameworks • Einführung JavaEE-Plattform • Ressourcen-Management und Services von Applikationsservern und Containern • Serverseitige Komponenten, verteilte Objektsysteme, Objektpersistenz, Messaging • Design Patterns • Zugriffskontrolle und Sicherheit | | | | | |
| Literatur Szyperski: Component Software, Pearson, 2002 Ihns et al.: EJB3 professionell, dpunkt, 2007 Walls: Spring im Einsatz, Hanser, 2008 Wütherich: Die OSGI Service-Plattform, dpunkt, 2008 | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Folien, Projektaufgabe | | | | | |

| | | | | | |
|---|--|-------------|--|--|--|
| Modulbezeichnung / Prüfungsfach Grafisch-Interaktive Systeme | | Liste SB | Modulnummer 7600 | Sem. 4 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 10 cp, 300 h 6 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (4 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt) | | | Dozenten Dörner, Schulz, Schwanecke | | Verantwortlich Schulz |
| Inhaltliche Voraussetzungen • Objektorientierte Softwareentwicklung: Programmierkenntnisse • Computergrafik: Grundkenntnisse der generativen Computergrafik und über GDV-Systeme | | | | | |
| Lernziele Das Gebiet der Computergrafik befindet sich nach wie vor in einer sehr dynamischen Entwicklung. Interaktive Grafiksysteme spielen hierbei eine besonders große Rolle: Auf Grund der gestiegenen Hardwareleistung kann 3D-Grafik zunehmend zur Gestaltung anspruchsvoller Benutzerschnittstellen herangezogen werden. Ziel dieses Moduls ist es, die Studierenden ausführlich in Themen aus diesen Gebieten einzuführen und durch ein größeres Projekt zu eigener Entwicklungsarbeit zu befähigen. Dabei ist die Mitwirkung der Bachelor-Studierenden auch an Masterarbeiten und an Projekten im Rahmen des Doktorandenkollegs vorgesehen. | | | | | |
| Inhalte Dieser Modul, der sich auf künftige aktuelle Entwicklungen bezieht, kann naturgemäß keine vorab festgelegten Lehrinhalte haben. | | | | | |
| Literatur Ausgewählte Originalliteratur | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter (als pdf-Dateien) | | | | | |

| | | | | | |
|---|--|-------------|--------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Mobile Computing | | Liste SB | Modulnummer 7610 | Sem. 4 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 10 cp, 300 h 6 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (4 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt) | | | Dozenten Gergeleit | | Verantwortlich Gergeleit |
| Inhaltliche Voraussetzungen • Modul "Rechnernetze und Telekommunikation": Grundkenntnisse über die Vernetzung von Rechnern | | | | | |
| <p>Lernziele</p> <p>Während die Vergangenheit vorwiegend durch stationäre Computer geprägt war, sind die persönlichen Computer zunehmend mobile Laptops, Netbooks und Smartphones. Auch im industriellen Einsatz nehmen mobile Endgeräte einen immer wichtigeren Platz ein. Diese Geräte sind durch verschiedene drahtlose Netzwerktechnologien umfassend vernetzt. Aber obwohl die Leistung der mobilen Endgeräte und Netze immer weiter zunimmt, gibt es auch in Zukunft wesentliche Aspekte, die beim Entwurf und Betrieb von Mobile Computing Systemen speziell berücksichtigt werden müssen. In diesem Modul:</p> <ul style="list-style-type: none"> • Lernen die Studierenden die grundlegenden Begriffe, Technologien und Randbedingungen der drahtlosen Kommunikation und der mobilen Endgeräte kennen. • Sie sind in der Lage, für gegebene Probleme eine geeignete Kommunikations- und Geräteplattform auszuwählen, bestehende Systeme zu bewerten und selbst mobile Netzwerke zu planen. • Lernen die Studierenden im Praktikum mit drahtlosen Netzwerk-Komponenten und Mobilien Endgeräten selbst Mobile Netze zu konfigurieren, zu monitoren, zu vermessen und spezifische Aspekte in einfachen Anwendungen zu erproben. <p>Neben dem Erwerb dieser spezifischen Analyse-, Design- und Realisierungskompetenzen haben sie ihre kommunikative Kompetenz durch Präsentation von eigenen Projektergebnissen und ihre sozialen Kompetenzen durch Arbeit in kleinen Projektteams vertieft.</p> | | | | | |
| <p>Inhalte</p> <ul style="list-style-type: none"> • Grundlagen der drahtlosen Kommunikation • Mobilfunknetze (2G und 3G) • drahtlose lokale Netze (IEEE 802.11, Bluetooth) • drahtlose Access-Networks (WiMAX) • Simulation und Vermessung von Funkausleuchtungen • Frequenzmanagement • Protokolle zur mobilen und spontanen Vernetzung • Mobile IP • Dienstgüte in mobilen Netzen • mobile Endgeräte • SW-Plattformen und höhere Dienste • Sicherheit der mobilen Netze und der von mobilen Endgeräten • aktuelle Entwicklungen (z.B. MiMo, LTE) | | | | | |
| <p>Literatur</p> <p>Schiller: Mobilkommunikation, Pearson Studium, 2003</p> | | | | | |
| <p>Medienform / Unterrichtssprache</p> <ul style="list-style-type: none"> • Veranstaltungsspezifische Web-Seite • Skript, Vorlesungsfolien und Übungsblätter | | | | | |

| | | | | | |
|--|--|-------------|--|--|--|
| Modulbezeichnung / Prüfungsfach Project - Current Topics in Applied Computer Science | | Liste SB | Modulnummer 7620 | Sem. 4 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 10 cp, 300 h 6 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (4 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt) | | | Dozenten Dozenten des Studienbereichs, Gastprofessoren, Lehrbeauftragte | | Verantwortlich Gergeleit |
| Inhaltliche Voraussetzungen | | | | | |
| Lernziele Changing requirements demand a flexible response while still developing working solutions is typical in a professional environment. • work independently and gain relevant skills/knowledge • develop a practical solution to a demanding problem at hand | | | | | |
| Inhalte • Guided exploration of a current topic in applied computer science • Self-reliant execution of a project with a thematical focus in small teams • Development of a working application | | | | | |
| Literatur depending on topic | | | | | |
| Medienform / Unterrichtssprache • depending on topic • course language is English | | | | | |

| | | | | | |
|---|--|-------------|--|--|--|
| Modulbezeichnung / Prüfungsfach Projekt zu aktuellen Themen der Angewandten Informatik | | Liste SB | Modulnummer 7630 | Sem. 4 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 10 cp, 300 h 6 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (4 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt) | | | Dozenten Dozenten des Studienbereichs | | Verantwortlich Gergeleit |
| Inhaltliche Voraussetzungen | | | | | |
| Lernziele Flexibel auf neue Anforderungen zu reagieren und selbständig Lösungen zu entwickeln ist typisch im Berufsleben • Selbständiges Erarbeiten von Lösungen • Praktische Umsetzung einer anspruchsvollen Aufgabenstellung | | | | | |
| Inhalte • Begleitete Erarbeitung von aktuellen Themen der angewandten Informatik • Eigenständige Durchführung von Projekten mit spezifischen Themenschwerpunkt in kleinen Gruppen. • Praktische Realisierung einer funktionierenden Anwendung. | | | | | |
| Literatur Literatur passend zum gewählten Schwerpunkt | | | | | |
| Medienform / Unterrichtssprache • Medienform passend zum gewählten Schwerpunkt | | | | | |

| | | | | | |
|--|--|-------------|-------------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Service-orientierte Architekturen | | Liste SB | Modulnummer 7640 | Sem. 4 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 10 cp, 300 h 6 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (4 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt) | | | Dozenten Kröger, Weitz, Werntges | | Verantwortlich Kröger |
| Inhaltliche Voraussetzungen • Modul Programmiermethodik und Programmiertechniken: Programmieren in Java • Modul Betriebssysteme und Verteilte Systeme: Grundlagen der Verteilten Systeme | | | | | |
| Lernziele Nach Abschluss des Moduls • kennen die Studierenden Service-orientierte Architekturen und Design-Patterns zu deren Bildung und können diese bewerten, • können sie wiederverwendbare Dienste aus Geschäftsprozessen ableiten, • kennen sie Architekturen zur Bildung von Service-Verbänden aus Diensten und können diese anwenden, • sind sie in der Lage, entsprechende Architekturen auf der Basis von Web Services zu realisieren, • kennen sie Methoden zur Instrumentierung von Services und können diese zur Gewinnung von Performance-Kenndaten einsetzen. Der Modul erweitert die technologischen Kompetenzen, erweitert die fachübergreifenden Kompetenzen in Hinblick auf betriebswirtschaftliche Zusammenhänge, vertieft die Design- und Realisierungskompetenzen in Hinblick auf die Nutzung aktueller Technologien und fördert die kommunikativen Kompetenzen sowie die Projektmanagement-Kompetenz durch Projektarbeit. | | | | | |
| Inhalte • Einführung und Terminologie • Service-Orientierung und Abstraktion • Service Contracts • Zustandsmanagement • Message-Exchange Patterns • Service Discovery • Beispiel: Web Services • Service Composition • Wiederverwendbarkeit von Services • Beispiel: Service Component Architecture • Instrumentierung und Monitoring • Service Management und Selbstmanagement • Geschäftsprozess-Management • SOA Governance • Frameworks • Praktikum mit geeigneten SOA-Plattformen | | | | | |
| Literatur Th. Erl: SOA Principles of Service Design, Prentice Hall, 2008 N. Josuttis: SOA in der Praxis: System-Design für verteilte Geschäftsprozesse, dpunkt.verlag, 2008 | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifischen Web-Seite • Skript, Praktikumsaufgaben, Projektbeschreibungen als PDF | | | | | |

| | | | | | |
|--|--|-------------|--------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Sichere Systeme | | Liste SB | Modulnummer 7650 | Sem. 4 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 10 cp, 300 h 6 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (4 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt) | | | Dozenten Geib, Reith | | Verantwortlich Geib |
| Inhaltliche Voraussetzungen • Security: Grundlagen der Informationssicherheit • Fehlertolerante Systeme: Grundlagen der funktionalen Sicherheit (Safety) • Rechnernetze und Telekommunikation: Protokolle und Dienste | | | | | |
| Lernziele Studierende mit den grundlegenden Anliegen, Konzepten und Begriffen der Informationssicherheit vertraut machen und einen strukturierten Einstieg in aktuelle Themenstellungen bei der Realisierung von sicheren IT-Systemen geben. • Gewinnung eines grundlegenden Überblicks über alternative Vorgehensweisen zur Erstellung von Sicherheitskonzepten für informationstechnische Systeme • Vermittlung des Wissens über technische Schutzmaßnahmen und Sicherheitskomponenten, die in gegenwärtigen IT-Systemen eingesetzt werden • Eine konkrete Problemstellung aus dem Aufgabengebiet eigenständig bearbeiten | | | | | |
| Inhalte • Einführung (Anliegen, Konzepte und Begriffsdefinitionen) • Risiken und Sicherheitsanforderungen (Sicherheitsrisiken und deren Überführung in konkrete Sicherheitsanforderungen) • Erstellung von Sicherheitskonzepten (Erkennung, Bewertung und Begegnung von Sicherheitsrisiken, Durchführung von Risikoanalysen) • Basismechanismen der Kryptographie (elementare Moduloarithmetik, symmetrische und asymmetrische Kryptomechanismen und ihre Wirkung, Konzept der Einwegfunktion und daraus abgeleiteter Hashfunktionen, Behandlung des Schlüsselaustauschproblems) • Grundlagen der Zuverlässigkeitstechnik (Fehlertoleranz, Verfügbarkeit und Lebensdauer) • Rechnersicherheit (unterschiedliche Identifikations- und Authentifikationsverfahren, wesentliche Zugriffskontrollmethoden, Einsatz von Beweissicherung und Protokollierung, Architekturmaßnahmen und Methoden auf der Hardwareebene sowie im Betriebssystem) • Netzwerksicherheit (Risiken vernetzter Systeme, Problematik der Zugriffskontrolle in Netzen, Firewall-Prinzipien und -Architekturen, besondere Risiken bei Netzübergängen) • Websicherheit (Sicherheitsfunktionen in Betriebssystemen, Web-Servern und Datenbanken) • Sicherheitswerkzeuge und Bewertung der Sicherheit (Prüf-, Test- und Diagnosetechniken) | | | | | |
| Literatur Bruce Schneier: Applied Cryptography, 1996 Martin Raeppele: Sicherheitskonzepte für das Internet, dpunkt.verlag, 2001 Claudia Eckert: IT-Sicherheit, Oldenbourg-Verlag, 2008 Schneeweis, W.G.: Zuverlässigkeitstechnik, Datakontext-Verlag, 1992 | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript, Folien und Übungsblätter | | | | | |

| | | | | | |
|---|--|-------------|--------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Simulationstechnik | | Liste SB | Modulnummer 7660 | Sem. 4 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 10 cp, 300 h 6 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (4 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt) | | | Dozenten Geib, Kröger | | Verantwortlich Geib |
| Inhaltliche Voraussetzungen • Diskrete Strukturen: Graphen und Algorithmen • Numerische Verfahren und Analysisgrundlagen: Integralrechnung, numerische Integration, Fehlerabschätzung • Statistik und Wahrscheinlichkeitsrechnung: Zufallsvariable, Zufallsprozess, Verteilungsfunktion und Erwartungswert • Modul Programmiermethodik und Programmiermethodiken: Programmieren in Java, teilweise C++ | | | | | |
| Lernziele Die Anforderungen an die Informations- und Kommunikationstechnik von heute führt zu hochkomplexen, meist vernetzten oder verteilten Anlagen, Systemen und Prozessen, deren Entwicklung ohne modellbasierte, computerunterstützte Methoden kaum mehr zu bewerkstelligen ist. Eine wichtige Methode stellt dabei die Simulation dar, deren effektive Anwendung allerdings ein breites Hintergrundwissen hinsichtlich einer Vielzahl von Teilsystemen aus unterschiedlichen technischen Disziplinen erfordert. Die Anwendungsschwerpunkte der Lehrveranstaltung decken die verschiedenen Phasen des Lifecycles vorwiegend automatisierter, mechatronischer Systeme ab, angefangen mit der Simulation einfacher Bewegungs- und Verhaltensabläufe in der Konzeptionsphase über die Simulation dynamischer Systemeigenschaften in der Entwurfs- und Implementierungsphase bis hin zur Echtzeitsimulation während der gesamten Betriebsphase. Nach erfolgreichem Abschluss der Lehrveranstaltung • verstehen die Studierenden Verfahren und Methoden zur Simulation dynamischer Systemeigenschaften, • können sie Simulationswerkzeuge und -hilfsmittel auswählen und anwenden, • können sie Simulationsprozesse steuern, durchführen und beurteilen, • können sie Simulationsergebnisse richtig interpretieren und deuten. Insgesamt trägt das Modul zu einer Erweiterung der technologischen Kompetenzen im Bereich der technischen Informatik bei, vertieft die Analyse- und Design-Kompetenzen und fördert die kommunikativen Kompetenzen sowie die Projektmanagement-Kompetenz durch Projektarbeit. | | | | | |
| Inhalte • Begrifflichkeiten und Grundlagen (Standards, Methoden, Technologien) • Diskrete ereignisgesteuerte Simulation • Wissenschaftliches Rechnen und Grundlagen der Systemdynamik • Entstehung von Simulationsmodellen (Modellbildungsprozess, Input, Phasen, Ergebnisse, Berechnungsverfahren und Analysetools) • Deterministische Simulationsmodelle im Bereich Automatisierungstechnik (z. B. Antriebsregelung) • Stochastische Simulationsmodelle im Bereich Netzwerktechnik (z. B. Warteschlangensysteme) • Echtzeitsimulation im Bereich Fertigungstechnik (z. B. Lagerverwaltung) • Architektur von Simulationsplattformen • Anwendung von MATLAB-Simulink und anderen Simulationswerkzeugen | | | | | |
| Literatur Hartmut Bossel: Systeme, Dynamik, Simulation - Modellbildung, Analyse und Simulation komplexer Systeme, Taschenbuch (2004) Uwe Kiencke: Ereignisdiskrete Systeme - Modellierung und Steuerung verteilter Systeme, Oldenbourg (2006) Eckehard Schnieder: Methoden der Automatisierungstechnik - Beschreibungsmittel, Modellkonzepte und Werkzeuge für Automatisierungssysteme, Vieweg Verlagsgesellschaft (1999) Josef Hoffmann: Matlab und Simulink - Beispielorientierte Einführung in die Simulation dynamischer Systeme, Addison-Wesley (1998) | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Folien und Übungsblätter als pdf | | | | | |

| | | | | | |
|--|--|-------------|--------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Systemprogrammierung | | Liste SB | Modulnummer 7670 | Sem. 4 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 10 cp, 300 h 6 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (4 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt) | | | Dozenten Kaiser | | Verantwortlich Kaiser |
| Inhaltliche Voraussetzungen • Modul "Betriebssysteme und Verteilte Systeme" oder z.B. das Buch Tanenbaum: "Moderne Betriebssysteme": Grundlagen der Betriebssysteme • Modul "Programmiermethodik und -techniken": Programmieren in C und Java | | | | | |
| Lernziele Systemprogrammierung bzw. systemnahe Programmierung bleibt ein Schlüssel zum Verständnis der internen Abläufe in einem IT-System. Vertiefte Kenntnisse dieser Systemebene befähigen die Studierenden Nebenläufigkeit und Verteilung in Anwendungen effizient zu nutzen, eigene Systemkomponenten entwickeln zu können und Fragen der technischen Systemsicherheit fundiert beantworten zu können. Nach erfolgreichem Abschluss dieser Lehrveranstaltung haben die Studierenden erworben: • vertiefte Kenntnisse über Betriebssysteme • vertieftes Wissen über die Systemaufrufe eines konkreten Systems (UNIX/Linux oder Windows) • Verständnis der Zusammenhänge, die die Ausführungen von Anwendungen in vielschichtig organisierten Rechensystemen ermöglichen • Fähigkeiten auch die internen Schnittstellen eines Betriebssystems zu nutzen, um eigene Komponenten (z.B. Treiber) integrieren zu können • die Kompetenz dieses Wissen dann selbstständig auf größere Aufgabenstellungen anwenden zu können (Praktikumsaufgaben) • kommunikative und soziale Kompetenzen durch die Präsentation der eigenen Arbeiten und das Arbeiten in kleinen Gruppen | | | | | |
| Inhalte • Grundlagen (Begriff der Systemprogrammierung, Betriebssysteme, Systemaufrufe, Systemprogrammierungssprachen, Beispiele, Portabilität, Programmierstil) • Konkrete Systeme (z.B. UNIX/Linux, Windows) • Ein- und Ausgabe, Directory-Handling • Speicherverwaltung • Prozess- und Thread-Manipulation • Interprozess-Kommunikation • Netzwerkprogrammierung • Treiber-Modelle und -Programmierung • Sicherheit und typ. Angriffstechniken • Wechselnde Inhalte der Projekte zu aktuellen Themen | | | | | |
| Literatur Weber: "Praktische Systemprogrammierung", Vieweg-Verlag, 1999 Love, Oram: "Linux System Programming", O'Reilly Media, 2007 Rubini, Corbet, Kroah-Hartman; "Linux Device Drivers", O'Reilly Media, 3. Auflage, 2005 | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript, Folien und Übungsblätter | | | | | |

| | | | | | |
|--|--|-------------|--------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Web-Engineering | | Liste SB | Modulnummer 7680 | Sem. 4 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 10 cp, 300 h 6 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (4 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 150 h Praktische Tätigkeit (Projekt) • 60 h Vor- und Nachbereitung (Theorieteil) • 90 h Anwesenheit (Vorlesung und Praktikum) | | | Dozenten Martin, Werntges | | Verantwortlich Martin |
| Inhaltliche Voraussetzungen • Webbasierte Anwendungen, Objekt-orientierte Software-Entwicklung, Softwaretechnik | | | | | |
| Lernziele Verständnis von aktuellen Konzepten, Methoden, Techniken, Werkzeugen und Erfahrungen zur ingenieurmäßigen Entwicklung von Web-Anwendungen sowie ihre praktische Anwendung in der eigenen Projektarbeit in Web-Entwicklerteams. • Bewerten von potentiellen Risiken von Web-Anwendungen • Befähigung, zukünftige Entwicklungen im Bereich des Web-Engineering zu verfolgen und zu beurteilen Die erworbenen Fähigkeiten tragen damit in besonderem Maße zur Ausprägung von spezifischen Analyse-, Design- und Realisierungskompetenzen und zur Erweiterung von spezifischen technologischen Kompetenzen bei. Teamorientierte Projektarbeit erhöht ferner Projektmanagement- und Selbst-Kompetenzen. | | | | | |
| Inhalte • Einführung in Web-Engineering (Motivation, Definition, Grundprinzipien) • Produktentwicklung • Requirements Engineering für Web-Anwendungen • Modellierung von Web-Anwendungen • Architektur von Web-Anwendungen • Testen von Web-Anwendungen • Web-Projektmanagement • Qualitätsaspekte (Usability, Performanz, Sicherheit) • Semantische Web-Anwendungen • Web-Frameworks • Mobile Applikationen • Mobile Endgeräte | | | | | |
| Literatur Kappel, Pröll, Reich, Teschitzegger: "Web-Engineering", dpunkt 2004 Pascal Hitzler: "Semantic Web", Springer 2008 Bernd Müller: "Java Server Faces", Hanser 2006 Ruby, Thomas, Heinemeier Hanssen: "Agile Web Development with Rails", The Pragmatic Bookshelf 2009 Leonard Richardson, Sam Ruby: "RESTful Web Services", O'Reilly 2007 | | | | | |
| Medienform / Unterrichtssprache • Web-Seite zur Veranstaltung • Folien, Übungsblätter | | | | | |

| | | | | | |
|---|--|-------------|--------------------------------|--|--|
| Modulbezeichnung / Prüfungsfach Graphentheorie und Graphenalgorithmen | | Liste SB | Modulnummer 7690 | Sem. 4 | Studiengang Angewandte Informatik (B.Sc.) |
| Credits / SWS 10 cp, 300 h 6 SWS | Lehrformen • Vorlesung (2 SWS) • Praktikum (4 SWS) | | Häufigkeit bei Nachfrage | Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL) | |
| Aufwand • 120 h Bearbeitung (Praktikumsaufgaben) • 60 h Vor- und Nachbereitung (Vorlesung, Seminar) • 120 h Anwesenheit (Vorlesung, Seminar, Praktikum) | | | Dozenten Reith | | Verantwortlich Reith |
| Inhaltliche Voraussetzungen • Modul "Diskrete Strukturen": Elementare Beweismethoden, Grundlagen von Relationen | | | | | |
| Lernziele Studierende mit den grundlegenden Konzepten und Begriffen der Graphentheorie vertraut machen. Kenntnisse von Algorithmen für Graphen. Anwendungsmöglichkeiten der Graphentheorie für praktische Aufgabenstellungen. • Gewinnung eines grundlegenden Überblicks über Anwendungen der Graphentheorie in der Informatik • Möglichkeiten und Grenzen von Graphenalgorithmen • Eine konkrete Problemstellung aus dem Aufgabengebiet eigenständig bearbeiten | | | | | |
| Inhalte • Grundbegriffe: Graph, Untergraph, Wege, Zusammenhang, Isomorphie, Bäume, gerichtete Graphen, bipartite Graphen, Darstellung von Graphen • Komplexität: effiziente Algorithmen und NP-Vollständigkeit • Suchen in Graphen: Tiefensuche, Breitensuche, Topologisches Sortieren • Kreis und Wege: Eulersche und Hamiltonische Kreise, kürzeste Wege, TSP • Bäume: Algorithmen für minimaler Spannäume • Planare Graphen • Färbungen: Vier-Farben-Satz • Flüsse: Algorithmus von Ford und Fulkerson, maximale Flüsse • Matchings: Grundlagen, Hochzeitsproblem | | | | | |
| Literatur S. Krumke und H. Noltemeier, Graphentheoretische Konzepte und Algorithmen, Teubner, 2005 R. Diestel, Graphentheorie, Springer, 2010 V. Turau, Algorithmische Graphentheorie, Oldenbourg, 2009 P. Tittmann, Graphentheorie: Eine anwendungsorientierte Einführung, Carl Hanser Verlag, 2011 | | | | | |
| Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Projektaufgabe in schriftlicher Form | | | | | |