

## Lehrinhalte der Masterstudien

**Informatik (921)**

**Netzwerke und Sicherheit (911)**

**Pervasive Computing (938)**

**Software Engineering (937)**

an der Technisch-Naturwissenschaftlichen Fakultät  
der Johannes Kepler Universität Linz

gültig ab WS 2010/11

**Studiendauer:** 4 Semester  
**Abschluss:** Dipl.-Ing.  
**Profil:** IT-Profis mit wissenschaftlicher Fundierung

### Fächer

	ECTS	Wochen- stunden
<i>Kernfach</i>		
Lehrveranstaltungen aus einem Masterprogramm	27,0	18
Seminar	3,0	2
Praktikum	7,5	5
<i>Nebenfach</i>		
Auswahl von Lehrveranstaltungen aus einem anderen Masterprogramm (ohne Praktikum)	18	12
<i>Wahlfächer</i>		
Informatik-Wahlfach	9,0	6
Freie Lehrveranstaltungen	9,0	6
<i>Masterarbeitsseminare</i>	12,0	6
<i>Masterarbeit</i>	30,0	
<i>Masterprüfung</i>	4,5	
<b>Gesamt</b>	<b>120,0</b>	<b>55</b>

# Semesteraufteilung

Std \ Sem	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
1.Sem	Kernfach												Freifach	Nebenfach							
2.Sem	Kernfach													Nebenfach							
3.Sem	Masterarbeit													Magister-arbeits-seminare				Wahlfach			
4.Sem	Masterarbeit													Magister-arbeits-seminare				Wahlfach			

## Lehrveranstaltungsarten

VO	Vorlesung (z.B. 2VO bedeutet 2-stündige Vorlesung)
UE	Übung
KV	Kombinierte Veranstaltung (d.h. Vorlesung + Übung)
PR	Praktikum
SE	Seminar

## Institute

BIO	Bioinformatik
CA	Computer Architecture
CG	Computer Graphics
CP	Computational Perception
FAW	Anwendungsorientierte Wissensverarbeitung
FGF	Frauen und Geschlechterforschung
FIM	Informationsverarbeitung und Mikroprozessortechnik
FMV	Formale Modelle und Verifikation
IIS	Institut Integriert Studieren
PC	Pervasive Computing
RIIC	Integrierte Schaltungen
RISC	Symbolisches Rechnen
SEA	Systems Engineering und Automation
SSW	Systemsoftware
TK	Telekooperation

# Masterstudium Informatik (921)

## Wintersemester

### Model Checking (2VO + 1UE; FMV, Biere)

*Goals:* Understanding algorithms and data structures for the analysis of state-based formal models and their implementation.

*Contents:* Verification vs Synthesis, Simulation of Automata, Abstraction and Refinement, Bisimulation, Automata Minimization, Explicit State Reachability Analysis, State Hashing, Partial Order Reduction, Fair Cycle Detection, Binary Decisions Diagrams (BDDs), And-Inverter Graphs (AIGs).

### Software-Architekturen (3KV; SSW, SWE, Blaschek – Weinreich)

*Ziele:* Kenntnis der wichtigsten Architekturmodelle zur Konstruktion umfangreicher Softwaresysteme. Vermittlung der Fähigkeit, für konkrete Anwendungsfälle geeignete Architekturen auszuwählen, sie softwaretechnisch umzusetzen, kritisch zu beurteilen und zu dokumentieren.

*Inhalt:* Grundlagen, Entwurststile, Strukturkomplexität, Modularisierung, Dokumentation, Objektorientierte Architekturen, Komponentenbasierte Architekturen, Architekturen verteilter Systeme, Serviceorientierte Architekturen.

*Übungen:* Schriftliche Aufgaben zu ausgewählten Themen. Diskussion der Themen in den Übungsblöcken der Lehrveranstaltung.

### Parallel Computing (3KV; CA, Strumpfen)

*Goals:* This course introduces basic parallel algorithms and design techniques of parallel computing.

*Contents:* The course emphasizes the practical aspects of multithreaded programming of multicores and GPGPU's. Two programming models are introduced: Cilk for divide-and-conquer style parallel algorithms, and Cuda for data-parallel algorithms. The course teaches design and analysis of parallel algorithms and programming practices for problems including matrix computations, sorting, searching, string matching, graphs, computational geometry and others.

## Sommersemester

### Hardwareentwurf (2VO+1UE; RIIC, Hagelauer)

*Ziele:* Vertiefende Betrachtung der Entwurfsmethodik sowie der Herstellungsprozesse integrierter Schaltungen. Verstehen der Leistungsgrenzen heutiger Rechnersysteme. Funktionsprinzipien der Schnittstellen zwischen analoger und digitaler Welt.

*Inhalt:* Aufbau und Betriebsbereiche des Feldeffekttransistors, CMOS-Schaltungstechnik und Berechnung, Realisierung von Grund- und Komplexgattern. Entwurfsregeln bei digitalen Schaltungen, Limitierende Faktoren (maximale Taktfrequenzen, etc.), Was sind ASICs?, ASIC-Typen und deren Aufbau und Funktion (RAM, ROM, etc.), Entwurfsablauf und Chipdesignmethoden, ADC und DAC, Fertigungstechniken integrierter Schaltungen.

*Übungen:* Entwurf digitaler Schaltungen und Berechnung wesentlicher Parameter, Realisierung von Algorithmen (in Form eines Ablaufdiagramms) als digitale Schaltung, Abschätzung von Designfehlern, Verstehen der Auswirkung von Fertigungstoleranzen anhand kleiner Rechenbeispiele.

## **Cooperative Systems (2VO+1UE; TK, Kotsis - Khalil)**

*Goals:* Inspired by the cooperative nature of human behavior, cooperative systems are built around multiple human and software entities, each, with limited resources and knowledge, that need to interact with each other to achieve common and/or individual goals within a specific context. The objectives of this course is to give students the technical knowledge, the theoretical background and the hand-on experience on cooperative systems from an algorithmic, functional, and architectural perspective.

*Contents:* In this course, students will get a deep insight experience on designing cooperative systems (user centered design, design patterns, design of general community support, e.g. registration, guidance, protection, awareness) as well as communication, coordination, and collaboration principles (basics of communication theory, electronic data interchange, supporting textual communication, rich media communication, group communication, creating places for collaboration, handling collaborative sessions, working with shared material, ensuring consistency and synchronization, coordination models, and coordination methods). A specific focus will be on agent-based cooperative systems which is one of the most important areas to have emerged in information technology and have been used in an increasingly wide variety of applications, ranging from comparatively small systems like email filters to large, open, complex, mission critical systems as air traffic control.

*Lab:* The objectives of this lab part of the course is to give students a hand-on experience in building cooperative systems using the Java Agent DEvelopment Framework (JADE) which is a software framework to develop agent-based applications in compliance with the FIPA specifications for interoperable intelligent multi-agent systems. The course is complemented by exercises.

## **Knowledge-centered Systems (3KV; FAW, Küng – Pröll - WöB)**

*Ziele:* Grundlegendes Verständnis für Knowledge-Centered Systems mit den Schwerpunkten Information Retrieval, Wissensbasierte Systeme und Integrierte Informationssysteme. Information und Wissen stellen einen ganz wesentlichen Produktionsfaktor in unterschiedlichen Bereichen - von produzierenden Unternehmen bis Life Science Anwendungen - dar. Information Retrieval Systeme, Wissensbasierte Systeme und hoch integrierte Informationssysteme stellen in diesem Zusammenhang unterschiedliche technologische Paradigmen zur persistenten Haltung, zur Verarbeitung und zum Wiedergewinnen von Information bzw. Wissen bereit. Es werden Grundlagen zu diesen Themen vermittelt und deren Anwendungen gezeigt.

*Inhalt:* Grundlagen und erweiterte Konzepte des klassischen Information Retrieval (Recall/Precision, Dokument-Representation, IR-Modelle, Thesaurus, Klassifikation, Relevance Feedback, String Similarity) und Anwendungen des Information Retrieval (Information Filtering, Recommender Systeme, Digital Libraries, Information Extraction, IR und Datenbanken); erweiterte Basiskonzepte von Wissensbasierten Systemen und Semantischen Netzen, Prolog, Datalog, Data Mining und Datawarehouses; Informationsintegration (Verteilung, Autonomie, Heterogenität, Architekturen, Semantische Integration), Geschäftsprozessmanagement und Workflow-Managementsysteme, Computer Integrated Manufacturing (CIM), Enterprise Application Integration (EAI), Data Warehousing, Semantic Information Management & Data Grids (RDF, OWL, SWRL, Storing RDF, Triple Stores, Reasoning, G-SDAM)

*Übungen:* Übungen zu den oben genannten theoretischen Grundlagen und Anwendungen (inkludiert Programmieraufgaben in Prolog und Datalog) sowie Themenpräsentationen durch die Studierenden.

# Masterstudium Netzwerke u. Sicherheit (911)

## Wintersemester

### Einführung IT-Sicherheit (2VO; FIM, Mühlbacher)

*Ziele:* Zentrale Begriffe wie IT-System, Sicherheit, Risiko werden erläutert: Sicherheitsfragen sind nicht nur technisch, sondern auch strategisch und organisatorisch zu sehen. Neben einem Überblick über Malware stehen Sicherheitsmodelle im Mittelpunkt.

*Inhalt:* Schutzziele bei IT-Sicherheit: Authentizität, Datenintegrität, Vertraulichkeit, Verfügbarkeit, Verbindlichkeit, Anonymität/Pseudonymität; Bedrohungen und Risiko, Malware; Identifizierung, Authentisierung, Autorisierung; Kryptografische Hashfunktionen sowie deren Eigenschaften und Komplexität, Passwortspeicherung, Challenge Response-Verfahren; Zugriffsmodelle, Zugriffskontrollstrategien: DAC, MAC, RBAC, Fallstudien; Einzelvorträge von Praktikern zu ausgewählten Themen.

### Informationssicherheitsmanagement (1VO; FIM, Schaumüller-Bichl)

*Ziele:* Informationen und die sie verarbeitenden Systeme und Prozesse sind wichtige Werte jeder Organisation. Informationssicherheitsmanagement (ISM) soll die Vertraulichkeit, Integrität und Verfügbarkeit von Informationen sicherstellen. Die Vorlesung stellt die wichtigsten Vorgehensweisen und Standards auf dem Gebiet des Informationssicherheitsmanagements vor und zeigt Lösungswege für die Praxis auf.

*Inhalt:* Das Österreichische Sicherheitshandbuch: Sicherheitsmanagementprozess, Sicherheitspolitik, Risikoanalysestrategien, Info SiG, Info SiV, Industrial Security, Akkreditierung. Die ISO/IEC 27000 Familie, Zertifizierung, Risikomanagement, Sicherheitskennzahlen, Standards. IT-Grundschutz des BSI 100-1, 100-2, 100-3, GSTool, Zertifizierung. Cobit, Common Criteria.

### IT-Recht und Computerforensik (2 VO; FIM, Sonntag)

*Ziele:* Verständnis der grundlegenden Konzepte der Computerforensik, ihres Spannungsfeldes zwischen Erlaubtem und Verbotenem, ihrer technischen Grenzen, sowie des zugehörigen Rechtsrahmens. Praktische Erfahrung mit ausgewählten Problemen der Beweissicherung sowie der Extraktion, Beobachtung, bzw. Wiederherstellung von Daten aus verschiedenen Quellen.

*Inhalt:* Im rechtlichen Teil werden Vorratsdatenspeicherung und Abhören bzw. Durchsuchungen/Beschlagnahmen erläutert. Weiters werden die Regelungen für Internetnutzung am Arbeitsplatz und der Schutz von Zugangskontrollsystemen (Hard- und Software) besprochen. Ebenso enthalten sind der Datenschutz, der detailliert untersucht wird, sowie ein Überblick über strafrechtliche Regelungen im Computerbereich, z.B. Datenbeschädigung und Computerbetrug. Der technische Teil beschäftigt sich u.a. mit der Beweissicherung (Images von Festplatten sowie Sicherung dynamischer Daten zur Laufzeit), der Analyse von Dateisystemen (auffinden bzw. wiederherstellen gelöschter Daten), der Rekonstruktion von Webbrowsing- und E-Mail-Aktivitäten, sowie der Rückverfolgung von E-Mails. Ein weiterer Teil ist die Analyse des Netzwerkverkehrs, sowohl anhand von Aufzeichnungen wie auch live.

### Systemadministration (2KV; FIM, Hörmannseder - Putzinger)

*Ziele:* Verständnis für den Aufbau einer stabilen Server-Infrastruktur inkl. Aufsetzen und Konfiguration elementarer Dienste unter Einbeziehungen von Überlegungen in Richtung Verfügbarkeit und sicherer Rechtevergabe.

*Inhalt:* Server-Administration auf Basis Windows / Linux; Virtualisierung, Verzeichnisdienste (ADS, LDAP), Fileserver, RAID, Printserver, DHCP, DNS, HTTP, FTP, Email + AntiSpam + Content-Scanning; Backup+Recovery; System-Management; Clustering.

*Übungen:* Praktische Vertiefung der Kenntnisse und Tests an Hand einzelner Standardbeispiele; Ausarbeitung eines Beispiel-Szenarios inkl. Präsentation der Ergebnisse.

### Parallel Computing (3KV; CA, Strumpfen)

*Siehe Masterstudium Informatik*

## **Sommersemester**

### **Netzwerk-Management** (3KV; FIM, Hörmannseder)

*Ziele:* Verständnis für die Konzeption, den Aufbau und den Betrieb von Netzwerken auf Layer-2 "Switching", Layer-3 "Routing" und Layer-4 bzw. 7 "Load Balancing" des ISO/OSI-7-Schichten-Modells. Dabei wird insbesondere Wert auf Sicherheits- und Verfügbarkeitsaspekte sowie auf Überwachung gelegt. Damit stehen auch Fehlererkennung, Protokollierung, Performance-Management, zentrale netzwerkweite Authentifizierung sowie Sicherheitskomponenten im Vordergrund.

*Inhalt:* Aufgabe und Einsatz von Netzwerk-Management-Systemen, VLANs, Spanning Tree, Link Aggregation, 802.1x, Radius, SNMP, Routingprotokolle (z.B. OSPF), VRRP, QoS, Load balancing, VPNs, IDS/IPS, Firewall-Systeme, Security-Scanner.

*Übungen:* Aufbau und Konfiguration von Beispiel-Szenarien inkl. Sicherheitsüberprüfung; Durchführung einzelner Netzwerkattacken in ausgewählten Beispielkonfigurationen inkl. Überlegungen zur Erkennung und für Gegenmaßnahmen.

Die Lehrveranstaltung setzt "Netzwerke und verteilte Systeme" voraus, baut auf "Einführung IT-Sicherheit" sowie "Systemadministration" auf und arbeitet in enger fachlicher Abstimmung mit der Lehrveranstaltung "Kryptographie".

### **Kryptographie** (3KV; CP, Scharinger)

*Ziele:* Solides Wissen über kryptographische Methoden zur Daten- und Kommunikationssicherheit ist in der heutigen Zeit von enormer Bedeutung. In dieser LVA soll daher versucht werden, interessierte Studenten in einer umfassenden Form mit den aktuellen Konzepten, Algorithmen, Protokollen und Systemen, die in der modernen Informatik, Kommunikations- und Informationstechnik Verwendung finden, vertraut zumachen.

*Inhalt:* Grundlagen der Kryptographie; Symmetrische Blockchiffren (DES, IDEA, SKIPJACK, AES); Symmetrische Stromchiffren (Schieberegister, RC4, SEAL, A5), Asymmetrische Kryptosysteme (DH, ElGamal, RSA); Digitale Signaturen; Kryptographische Hashfunktionen; ECC (Kryptographie mit elliptischen Kurven); Anwendungen der Kryptographie (PGP - Pretty Good Privacy, SSL/TLS - Secure Sockets-Layer, SET und 3D-Secure).

### **Sicherheitsmodelle in Informationssystemen** (2KV; FAW, Küng)

*Ziele:* Richtig implementierte Sicherheit in Informationssystemen muss die Interessen der Benutzer aber auch der in sonstiger Weise Betroffenen bezüglich Verfügbarkeit, Vertraulichkeit, Anonymität, Integrität und gewährleisten. Dabei sind insbesondere die Zugriffskontrolle sowie die Datenfluss- und Inferenzkontrolle von Bedeutung. Ziel dieser Lehrveranstaltung ist eine umfassende Vermittlung von Modellen, Konzepten und Architekturen, die eine solche "richtige Implementierung von Sicherheit" in Informationssystemen unterstützen.

*Inhalt:* Die Lehrveranstaltung enthält im ersten Teil eine Wiederholung der generellen Architekturen von Informationssystemen mit speziellem Fokus auf sicherheitsrelevante Teile. Der zweite Teil widmet sich ausführlich den aus der Theorie bekannten Sicherheitsmodellen und -konzepten für Informationssysteme. Im dritten Teil wird dann gezeigt, inwieweit diese in den derzeit aktuellen Datenbankmanagementsystemen (u.a. ORACLE, SQL-Server, DB2, MySQL) sowie in den unterschiedlichen Architekturen und Frameworks für komplexere Informationssysteme (u.a. verteilte Informationssysteme incl. Datawarehouses, ERP-Systeme, Web-basierte Informationssysteme) umgesetzt werden.

*Übungen:* In den Übungen sind konkrete Sicherheitsmodelle in DBMS zu implementieren, sowie komplexere Systeme zu konzipieren, die einem bestimmten, vorgegebenen Sicherheitskonzept entsprechen.

# Masterstudium Pervasive Computing (938)

## *Wintersemester*

### **Pervasive Computing Infrastructure** (2VO+1UE; PC, Ferscha)

*Goals:* The continuously unfolding research field of Pervasive and Ubiquitous Computing will be presented based on the key research issues and contributions, the principle methods of system design and development, the state-of-the art of technology and the potential domains of application.

*Contents:* Historical roots, vision and paradigms; enabling technologies for identification, positioning, localization and tracking; wireless communication, sensor/actuator systems, system architectures for networked embedded systems; systems software and middleware, system design for context-awareness, coordination, unobtrusiveness and multimodal interaction; ethnographic, social, legal, security and privacy issues.

*Practical Track:* Students will be involved at all stages of system design and development, earning hands on-experience in exploratory research methods.

### **Pervasive Computing Systems Development** (2VO+1UE; PC, external lecturers, coordination: Ferscha)

*Goals:* The goal of this course is motivated by the fact that developing pervasive computing systems goes way beyond the development of software for traditional computing systems, demanding e.g. abstractions for computing ensembles, real time, real space, goal orientedness, dependability, correctness, modalities of interaction, explicit and implicit use, usability and trust.

*Contents:* Design process models for pervasive computing systems; system models for adaptation, emergence, intelligence, ensembles and swarms; location, mobility and awareness models; software development based on programming abstractions for real time, location and space, context (multi-sensors), autonomy, self-management and self-organisation, emergent behaviour.

*Practical Track:* Students will be involved in focussed systems and software development projects in the application domains of wireless sensor networks; mobile and wearable computing; smart home/office/car/city; transport and logistics, sports and healthcare; learning, entertainment and gaming; etc.

### **Unconventional User Interaction** (2VO + 1UE; TK, external lecturers, coordination Kotsis)

*Goals:* Keyboard, mouse and monitor have been the typical I/O devices following the personal computing paradigm and strongly influencing our habits on how to interact with computers. Technological advances (visual computing, sensors, speech analysis and synthesis, ambient displays, etc.) are opening up a plethora of new interaction possibilities which are to be presented and discussed in this lecture in a systematic way. Basic interaction principles will be presented along with their possible implementation in hard- and software.

*Contents:* Classification of UIs, interaction principles, selected I/O technologies, development methods and tools, evaluation.

*Practical Track:* Students will work in small project teams on the prototypical implementation of UUI solutions. The first subproject will be dedicated towards UUI design, the second will focus on UUI evaluation.

## ***Sommersemester***

### **Machine Learning and Pattern Classification** (3KV; CP, Widmer)

*Goals:* To give an overview of standard methods in the field of pattern classification, machine learning, and statistical data modelling. To explain the basic concepts and methods in the field, and demonstrate the application of these methods in a variety of complex tasks.

*Contents:* Bayes classification and Bayes error; density estimation; nearest-neighbour classification; standard classifiers in machine learning (decision trees, rules, Naive Bayes, feed-forward neural networks, support vector machines); empirical evaluation of classifiers; clustering and (Gaussian) mixture models; dimensionality reduction and data projection methods; Markov processes and Hidden Markov Models.

*Practical Track:* Students will carry out a pattern classification project of real-world complexity in several stages, from feature definition and extraction to the training of various classifiers and systematic experimentation.

### **Cooperative Systems** (2VO+1UE; TK, Kotsis)

*Siehe Masterstudium Informatik*

### **Mixed Reality Systems** (3KV; CG; Anthes)

*Goals:* This course teaches concepts, methodologies and usage of tools, which can be used to create virtual worlds and interconnect them with the real world. Students should be enabled to develop their own mixed reality applications from both augmented and virtual reality.

*Contents:* Concepts and applications of mixed reality, position tracking, mixed reality display technologies, scene graph programming, interaction and navigation techniques, human-computer interfaces, input and output devices for virtual and augmented reality, programming such devices, examples and applications.

# Masterstudium Software Engineering (937)

## Wintersemester

### Formal Methods in Software Development (3KV; RISC, Schreiner)

*Ziele:* Verständnis der grundlegenden Konzepte der formalen Spezifikation von Programmen und Systemen sowie des formalen Schließens über die Korrektheit von Implementierungen in Bezug auf die Spezifikation. Praktischer Umgang mit diesen Prozess unterstützenden Software-Werkzeugen wie Beweis-Assistenten, statischen Checkern, und Model Checkern.

*Inhalt:* Die Rolle formaler Methoden in der Software-Entwicklung; die Spezifikation und Verifikation imperativer Programme (Hoare-Kalkül, der Computerunterstützte Beweis von Verifikationsbedingungen mit dem RISC ProofNavigator); die Spezifikation und Verifikation von Java Programmen (die Java Modeling Language JML, Extended Static Checking von Java/JML mit ESC/Java2, Verifikation von Java/JML mit KeY); die Spezifikation und Verifikation von nebenläufigen Systemen (Modellierung von Systemen, die Spezifikation von Systemeigenschaften in temporaler Logik, die Verifikation von Systemeigenschaften mit dem RISC ProofNavigator und dem Model Checker Spin).

*Übungen:* 3-4 Übungen zur formalen Spezifikation und Verifikation mit den oben genannten Software-Werkzeugen.

### Requirements Engineering (2KV; SEA, SWE, Grünbacher - Pomberger)

*Ziele:* Verstehen der Rolle der Anforderungen im Software Life Cycle. Kennenlernen verschiedener Methoden zur Ermittlung, Analyse, Verhandlung, Dokumentation und Validierung von Anforderungen. Praktische Anwendung von RE-Werkzeugen.

*Inhalt:* Aufgaben des RE, Rolle des RE im Software Life Cycle, Erhebungs- und Verhandlungstechniken (z.B. Kreativitätstechniken, Priorisierungstechniken, Moderationstechniken, Interviews, Dokumentenanalyse), prototypingbasierte Techniken, szenariobasierte Techniken, Use Case Analyse, IEEE Standard 830, Beschreibungsformen und Notationen (z.B. formale Spezifikationssprachen, Spezifikation von Anforderungen mit UML/SysML, User Stories), Struktur von Anforderungsdokumenten, Requirements- und Change Management, Requirements Traceability, Anforderungen und Software-Wiederverwendung, Rolle von Features und Anforderungen in Produktlinien.

*Übungen:* Anwendung der dargestellten Methoden und Werkzeuge anhand konkreter Anwendungsbeispiele, Workshops und Rollenspiele zur Verhandlung von Anforderungen, Use Case Analyse, Kennenlernen von Requirements Management Werkzeugen.

### Software-Architekturen (3KV; SSW, SWE, Blaschek – Weinreich)

*Siehe Masterstudium Informatik*

### Prinzipien von Programmiersprachen (3KV; SSW, Blaschek – Mössenböck - Prähofer)

*Ziele:* Breites Verständnis für Programmiersprachenparadigmen und -konzepte, insbesondere solche, die über Java hinausgehen. Diskussion dieser Konzepte an Hand ihrer Geschichte, ihrer Realisierungsalternativen in diversen Sprachen, der damit verbundenen Designüberlegungen, sowie ihrer Vor- und Nachteile. Kennenlernen ausgewählter Programmiersprachen.

*Inhalt:* Überblick (Klassifikation, Geschichte), Imperative Sprachen (Scoping, Typen, Ablaufkontrollstrukturen, Prozeduren und Parameter, Modularisierung, Parallelität, Ausnahmebehandlung, Spezielle Themen), Objektorientierte Sprachen (Begriffe und Mechanismen, reine objektorientierte Sprachen, Blöcke, Ko/Kontravarianz, Generizität, mehrfache Vererbung, Prototypen), Funktionale Sprachen (Lisp, Scheme, Haskell, XLinq, DLinq), Logische Sprachen (Unifikation, Prolog).

*Übungen:* Drei selbstgewählte Programmieraufgaben in einer imperativen, einer objektorientierten, einer funktionalen oder einer logischen Sprache.

## **Sommersemester**

### **Testen von Softwaresystemen** (2KV; SWE, Plösch - Sametinger)

*Ziele:* Verständnis für die Bedeutung des Testens im Softwareentwicklungsprozess und für die grundlegenden Konzepte und Techniken bei unterschiedlichen Testarten (z.B. Unit-Tests, Integrations- und Systemtest). Praktischer Umgang mit ausgewählten statischen und dynamischen Testwerkzeugen.

*Inhalt:* Prinzipien und Konzepte des Testens, Statische und dynamische Ansätze, Software-Qualität, Komponententests, Unit-Tests, Systemtests, Integrationstests, Akzeptanz-Tests, Performance-Tests, Release-Tests, Logging, Debugging, Test Management, Code-Inspektionen, Code Reviews, Walkthroughs, Software-Metriken, Ausgewählte Test-Werkzeuge.

*Übungen:* 3-4 Übungen zum Testen von Softwaresystem mit ausgewählten Testwerkzeugen.

### **Knowledge-centered Systems** (3KV; FAW, Küng – Pröll - WöB)

*Siehe Masterstudium Informatik*

### **Software-Prozesse und –Werkzeuge** (2KV; SEA, Grünbacher)

*Ziele:* Verstehen der Bedeutung von Software-Prozessen für die erfolgreiche Abwicklung komplexer Software-Entwicklungsprojekte. Die Fähigkeit, Software-Prozesse für einen konkreten Projekt- und Firmenkontext zu definieren oder bestehende Prozesse anzupassen. Kenntnis von Werkzeugen zur Unterstützung zentraler Software-Prozesse und von Methoden zur Prozessverbesserung

*Inhalt:* Gemeinsamkeiten und Unterschiede wesentlicher Entwicklungsparadigmen (Wasserfallmodell, Spiralmodell, Agile Entwicklung, ...), Prozessframeworks (ISO 15504, CMMI: Capability Maturity Model Integrated, ISO 15288), Prozessbeschreibungssprachen und Electronic Process Guides, Standards (V-Modell XT, Unified Process), Werkzeuge zur Unterstützung wichtiger Prozesse (z.B. Konfigurationsmanagement, Fehler- und Änderungsverfolgung, Build Werkzeuge), Prozessmessung und -verbesserung (GQM, Assessments, Retrospektiven), Value-based Software Engineering, Ausgewählte Software Prozesse (z.B. für Global Software Development, Wiederverwendung).

*Übungen:* Prozessbeschreibung mit dem Eclipse Process Framework, Tailoring von Prozessen, Kennenlernen ausgewählter Werkzeuge (z.B. Subversion, Bugzilla, Entwicklungsportale, Build Werkzeuge), Assessment Werkzeuge.

# Ausgewählte Lehrveranstaltungen für Informatik-Wahlfach

Nach Wahl

- 5 Stunden im Bachelorstudium
- 6 Stunden im Masterstudium

## **Advanced Model Checking** (2VO; FMV, Biere)

*Goal:* SAT and BDD based Symbolic Model Checking

*Contents:* SAT problem, SAT algorithms, DPLL, learning, SAT encoding, time frame expansion, bounded model checking, simple path constraints, image computation, interpolation, BDD based symbolic model checking

## **Agile Methoden der Softwareentwicklung** (2KV; SEA, Hoyer)

*Ziele:* Kennenlernen wichtiger Werte, Prinzipien und Methoden agiler Entwicklung. Praktische Umsetzung eines Entwicklungsprojekts im Team unter Verwendung wichtiger Praktiken des eXtreme Programming.

*Inhalt:* Agiles Manifest, Agile Methoden (Paarprogrammierung, testgetriebene Entwicklung, Refactoring, Story-Cards, schnelle Codereviews, etc.). Agile Prozesse (eXtreme Programming, Scrum, Crystal, Dynamic System Development Method, Feature Driven Development, etc.). Kriterien zur Eignung agiler vs. konventioneller Entwicklungsmethoden.

## **Anwendungsorientierte Wissensverarbeitung** (2VO; FAW, Küng)

*Ziele:* Einstieg in die Materie der Wissensbasierten Systeme und einige fundamentale Methoden im Detail.

*Inhalt:* Überblick über das Gebiet der Wissensbasierten Systeme, Prolog, Deduktive Datenbanken, Neuronale Netze, Semantische Technologien, Case based Reasoning.

## **Barrierefreie Systementwicklung** (2KV; IIS, Miesenberger)

*Ziele:* Bewusstseinsbildung über die Notwendigkeit der barrierefreien Gestaltung von Informationssystemen und Darstellung der speziellen Assistierenden Technologien (AT), die von Menschen mit Behinderungen an der Mensch-Maschine-Schnittstelle verwendet werden.

*Inhalt:* Begriff und Konzept "barrierefrei" und "Design for All", Potential der IKT und von AT für Menschen mit Behinderungen, ethische/politische/wirtschaftliche/soziale Bedeutung, Darstellung und Demonstration der AT für: - sehbehinderte und blinde - hörbehinderte und gehörlose - mobilitätsbehinderte - kognitiv beeinträchtigte Menschen

## **Biometrische Identifikation** (2VO; CP, Scharinger)

*Ziele:* Einführung in die Methoden der biometrischen Identifikation anhand diverser biometrischer Merkmale.

*Inhalt:* Grundlagen aus der Kryptographie, Grundlagen aus der digitalen Bildverarbeitung, Identifikationsmerkmale: Fingerabdruck, Gesicht, Ohren, Hand, Iris-Muster, Retina-Scan, Geruch, Gang von Personen, DNA, IR-Aufnahmen, Tastatur-Anschlag-Dynamik, Unterschrift, Sprache, Tests auf Lebendigkeit.

## **Computer Vision** (2VO; CG, Bimber)

*Goals:* Computer vision finds many applications in domains such as, 3D reconstruction, robotics, medical engineering, human-computer-interaction, contact free measurement, remote sensing, etc. This course will give first insights into the basics of computer vision.

*Contents:* Optical and geometric principles of cameras, projective imaging, geometric camera model and calibration, single- and multi-view geometry, estimation of disparities, 3D reconstruction from projective images and from motion, processing of images and range data: gradient domain processing, registration, segmentation, filtering, feature detection, texture analysis and synthesis, object recognition; applications such as computational photography.

## **Datenmodellierung und Applikationsentwicklung (2KV; FAW, WöB)**

*Ziel:* Mittels objektorientierter Methoden und Werkzeuge wird anhand von praxisorientierten Fallstudien in der Lehrveranstaltung gezeigt, welche Auswirkungen und welchen Nutzen die Datenmodellierung für die Entwicklung von Applikationen im Bereich der Informationssysteme hat. Moderne Entwicklungswerkzeuge kommen dabei zum Einsatz. Der Besuch der ergänzenden Lehrveranstaltung "Konzeptionelle Datenmodellierung" wird empfohlen.

*Inhalt:* Entwurf von Informationssystemen, Objektorientierte Datenmodelle, Unified Modeling Language (UML), Ontologien, Modellierungswerkzeuge

## **Debugging (2VO; FMV, Biere)**

*Goals:* Introduction to systematic Debugging

*Contents:* Failure and Defects, Analytic Debugging, Symbolic Debugging, Delta Debugging, Memory Leaks, Purify, Eraser, Profiling

## **Digitale Bildverarbeitung (2KV; CP, Scharinger)**

*Ziele:* Einführung in die Methoden und Systeme der Restauration, Verbesserung, Merkmals-Extraktion und Analyse digitaler Bilder.

*Inhalt:* Diskrete Transformationen, Restauration von Bildern, Geometrische Operationen, digitale Bildverbesserung, Segmentierung von Bildern in Regionen, Extraktion von Merkmalen, Erkennung und Klassifikation von Objekten, Kompression von Bilddaten mit Wavelets, JPEG-Standards, Digitale Wasserzeichen.

## **Digitale Sprachverarbeitung (2KV; CP, Scharinger)**

*Ziele:* Für 1D digitale Signale (Sprache, Musik) haben sich mit digitalen Filtern und der diskreten Fourier-Transformation mittlerweile klassische Werkzeuge etabliert, die eigentlich essentielles Wissen und Werkzeuge für jeden Techniker darstellen sollten. In dieser LVA wird Interessierten in einer selbsterklärenden Weise exakt dieses Wissen vermittelt, sodass dass der Student am Ende eine solide Grundlage für das Arbeiten mit digitalen Signalen (Sprache, Musik, etc.) besitzen sollte.

*Inhalt:* Phasoren, Digitalisierung, Feedforward- und Feedback Filter, Fourier-Transformation, z-Transformation, DFT und FFT, Fensterung von Signalen, Signalanalyse mittel FFT.

## **Embedded Systems (2KV; PC, Ferscha)**

*Ziele:* Die Lehrveranstaltung hat zum Ziel, die informationstechnologisch elementaren Fragestellungen im Embedded System Design und deren Implementierung aufzuzeigen. Neben methodischer Wissensvermittlung im Bereich des Embedded System Designs, in Technologiebewertung und Technologieauswahlkompetenz steht eine praktische Anwendung des erlernten Wissens im Vordergrund - dazu sind von den LVA-Teilnehmern mehrere Übungen im Fokusgebiet "Embedded Systems" in Einzel- bzw. Gruppenarbeiten auszuarbeiten.

*Inhalt:* Spezifikationssprachen und formale Modelle (VHDL, Statecharts, Petrinetze), Embedded Systems Software (Sprachen, Entwicklungsmethoden und -umgebungen, J2ME, Service Discovery, TinyOS und nesC), Globale Zeit und Uhrensynchronisation (Zeit und kausale Ordnung, Zeitmessung und Uhrensynchronisation), Echtzeit-Schedulingmechanismen, Hard- und Software-Codesign (Partitionierung, Co-Synthese, Codegenerierung), Embedded Systems Hardware (Digitale Signalverarbeitung, ASICs, Programmierbare Bausteine und Mikrokontroller, Digitale Signalprozessoren (DSPs), Logikbausteine (ASIPs, PLDs), Smartcards), Drahtlose Kommunikation (IrDA, RFID, IEEE 802.11, Bluetooth, ZigBee), Sensoren und Aktuatoren (elektromagnetische, optische, akustische und biometrische Sensoren, Mustererkennung, Aktuatoren).

## **Engineering of Software-intensive Systems (2KV, SEA, Egyed)**

*Goals:* Software-intensive Systems integrate software with hardware, middleware, firmware, and off-the-shelf ware. This lecture conveys the challenges associated with building such software-intensive systems where software is but a part of the system. It presents approaches and best practices for designing software-intensive systems and focuses on the early phases of the systems development life cycle (requirements, usage scenarios, and design).

*Contents:* The lectures discuss the unique role of software in software-intensive systems, top-down and bottom-up engineering, integration von components and advanced design concepts. The student is expected to design a software-intensive system of her choosing and explore design trade-offs.

### **Entwurf integrierter Schaltungen (2PR; RIIC, Ehrenstorfer)**

*Ziele:* Kennenlernen und Vertiefen der Hardwarebeschreibungssprache VHDL, Durchlaufen des FPGA-Designflows. Entwurf und Realisierung kleiner FPGA-basierter Projekte

*Inhalt:* Motivation für Hardwarebeschreibungssprachen, Syntax und Semantik von VHDL, Eventbasierter Simulationszyklus für digitale Schaltungen, synthesesfähiger VHDL-Code, Einbinden von spezifischen Funktionsblöcken eines FPGAs, Entwurf und Inbetriebnahme kleiner Projekte in einem FPGA-basierten Entwicklungsboard.

### **Gender Studies TNF-Einführung (2KV; FGF, Knoll)**

*Ziele:* Die Studierenden sollen nach der Lehrveranstaltung in der Lage sein, Bezüge zur eigenen Fachdisziplin herzustellen. Die Relevanz der Kategorie Gender in der Technik wird in und durch die Lehrveranstaltung auf unterschiedlichen Ebenen erkennbar und sichtbar gemacht.

*Inhalt:* Überblick über den aktuellen Stand der Debatte zu Gender in den Technik- und Ingenieurwissenschaften. Anhand ausgewählter Beispiele werden im Anschluss zentrale Forschungsfelder exemplarisch diskutiert. Die Lehrveranstaltung nähert sich der Thematik von drei Seiten her: (1) "Women in Science and Technology": strukturelle und symbolische Barrieren für Frauen in diesen Fachgebieten, Lösungsvorschläge zum Einstieg und zur Qualifikation. (2) "Science and Technology of Gender": Produktion und Determination von Vergeschlechtlichungen und Geschlechterdifferenzen in technologischen Artefakten. (3) "Gender in Science and Technology": metaanalytische Ansätze zur Forschungspraxis und Methodik der Technikwissenschaften, Mechanismen der Vergeschlechtlichung in technischen Erkenntnisinteressen, Theoriebildungen, Methoden, Interpretationen von Ergebnissen und Forschungspraxen.

### **Geschlecht und Wirtschaftsinformatik (2VO; FGF, Horvath)**

*Ziele:* Die Vorlesung ist Teil des Moduls „Basiskompetenz soziale und geschlechterspezifische Auswirkungen der IT“. Inhaltlich knüpft sie an die Themen der Vorlesung „Soziale Auswirkungen der IT“ an und stellt die jeweils geschlechtsspezifischen Aspekte in den Vordergrund. Studierende werden dadurch befähigt, geschlechtsspezifische Einflüsse und Auswirkungen der IT systematisch zu reflektieren und im eigenen Arbeitsumfeld zu berücksichtigen.

*Inhalt:* Die Vorlesung bietet eine Einführung in die wichtigsten Theorien zum Zusammenhang von Geschlecht und IT. Zunächst erfolgt eine Einführung in die wissenschaftliche Kategorie Geschlecht und zentrale theoretische Perspektiven. Anschließend werden verschiedene geschlechterrelevante Aspekte der IT theoretisch und empirisch beleuchtet, mit Themen wie Soziologische Theorien über Technik und IT, Geschlechterverhältnisse und sozialer Wandel in Zusammenhang mit IT, wechselseitige Einflüsse zwischen gesellschaftlichen Entwicklungen und IT, Sozialgeschichte der IT, gesellschaftliche Bilder der IT, soziale Gruppen und Zugang zu IT, Akzeptanz von IT.

### **Hardwareentwicklung mit programmierbarer Logik (2KV; RIIC, Ehrenstorfer)**

*Ziele:* Kennenlernen und Verstehen der Komponenten von Mikrocontrollern, Konzeption von Mikrocontroller-Applikationen, Programmierung von µCs und Ansteuerung typischer Hardwarekomponenten derartiger Systeme.

*Inhalt:* Aufbau und Funktion von Mikrocontrollern, Kennenlernen der unterschiedlichen µC-Komponenten, Programmierung derselbigen, Hardwaretechnische Ansteuerung von Mikrocontrollern, Anbindung und Programmierung typischer Peripheriebausteine, Fehlersuche und Debuggingmethoden.

### **Hardwareorientiertes Arbeiten an PCs (2PR; FIM, Bauer)**

*Ziele:* Das Ziel der LVA ist es, dass der Teilnehmer selbständig Hardwareinstallations- und Konfigurationsarbeiten am PC durchführen kann.

*Inhalt:* Einführung in die PC-Hardware, die wichtigsten Komponenten, das Innenleben des PC, Vorsichtsmaßnahmen, Systemkonfiguration, Massenspeicher (phys. u. log. Organisation v. Festplatten, Interface: ATA, SATA, SCSI), Betriebssysteminstallation (Boot- und Multibootfunktionen), I/O Schnittstellen (Seriell u. Parallel, USB, Netzwerkadapter), Erkennung und Lokalisierung von Defekten.

### **Information Displays (2VO; CG, Bimber)**

*Goals:* This course gives technical insights into present and future of display technologies and techniques. It describes how computer graphics, computer vision and display technology are linked, and deals with questions like: What will TVs be like in another 80 years from now?

*Contents:* Wave and geometric optics, holography, visual perception and display measures, basic display technologies, projection displays, projector-camera systems and techniques (including calibration and image correction), stereoscopic and auto-stereoscopic displays, computer generated holography, near-eye displays, real-time computer graphics and computer vision aspects for driving such displays.

## **Interactive Rendering and Visualization (2VO; CG, Bimber)**

*Goals:* This course will summarize the most important topics of interactive computer graphics and visualization. Participants will learn fundamental real-time rendering techniques for computer games, information visualization and scientific visualization.

*Contents:* Computational illumination, high-dynamic range visualization (including tone-mapping and inverse tone-mapping), non-photorealistic rendering (including cartoon-based rendering, artistic rendering, and sketch-based rendering), image-based rendering (including lightfields, billboards, and sprites), volume and point-based rendering, scalar-, vector- and tensor-visualization; domain modelling, table-, tree-, graph- and text-visualization.

## **Kapazitätsplanung (2KV; TK, Kotsis)**

*Ziele:* Verstehen der Methoden und Beherrschung der Werkzeuge zur Analyse der Leistung von Computersystemen und Netzwerken.

*Inhalt:* Grundlegende Leistungskennzahlen und Modelle. Methoden der Leistungsbewertung (modellierend und messend). Analysewerkzeuge. Fallstudien.

## **Konzeptionelle Datenmodellierung (2KV; FAW, WöB)**

*Ziele:* Die Lehrveranstaltung vermittelt ein theoretisches Fundament im Bereich der Datenmodellierung. Wesentliche Ziele sind das Verstehen des Zusammenwirkens von Daten- und Funktionsanalyse und der Einbettung von Datenanalyse und Modellierung im Rahmen einer modernen Anwendungsentwicklung. Der Besuch der ergänzenden Lehrveranstaltung "Datenmodellierung und Applikationsentwicklung" wird empfohlen.

*Inhalt:* Einführung in die Datenmodellierung, Entity Relationship-Modell und Erweiterungen, Semantische Datenmodelle

## **Logisches Programmieren (2KV; RISC, Kutsia)**

*Ziele:* Einführung in die Logische Programmierung. Verständnis der logischen Grundlagen und des Berechnungsmodells. Praktische Programmierung in Prolog lernen.

*Inhalt:* Logische Grundlagen, Horn Klausen, Einführung in die Prolog Programmierung, Datenstrukturen, wichtige Programmiertechniken in Prolog (Mapping, Rekursion, Akkumulatoren, Differenz-Strukturen), Rucksetzen (Backtracking), Erzeugen der mehrfachen Lösungen, Der Schnitt (the cut), ausgewählte Beispiele, Unifikation, Berechnungsmodell logischer Programme, Constraint-logic Programming, Grammatik Regeln (DCG), Effizienzfragen, Uebungen und Programmierprojekte.

## **Mensch-Maschine-Kommunikation (2VO; SSW, Blaschek)**

*Ziele:* Ergonomische Grundlagen und Elemente interaktiver Programme; Maßnahmen zur Förderung von Verständlichkeit und effizienter Bedienung.

*Inhalt:* Kommunikationstheorie, Wissen und Lernen, Hardware, Farbgestaltung, Layout, akustische Signale, Menüs, Fenster, Vermeidung von Zuständen, Entwurfstechniken, Ereignisorientierte Programmierung, Sicherheit und Robustheit.

## **Mobile Computing (2KV; TK, Khalil)**

*Goals:* The course aims at providing an interdisciplinary understanding of the concepts and issues underlying current developments in mobile communication systems and wireless computer networks.

*Content:* The course content will be organized under three areas: networking, services and applications. The course will cover Advances in Mobile Computing, Wireless Communications and Networks, Wireless Security, Mobile Services, Software Development, and Mobile Application Development

## **Model Engineering (2VO+1UE; BIO, Reschitzegger)**

*Ziele:* Erläuterung der grundlegenden Konzepte und Technologien der modellgetriebenen SW-Entwicklung.

*Inhalt:* Einführung in Model Engineering, Metamodellierung (MOF, UML2, Ecore/EMF), Modelltransformation (OCL, ATL als QVT-Realisierung), Codegenerierung (XML-basiert, Java-basiert, Modell-basiert), Grafische Modellierungsedatoren (Eclipse GMF), Neuerungen in UML2.

## **Modeling Internet Applications (2KV; TK, Schwinger)**

*Goals:* Goal is to provide an overview on the role of modelling in the development of internet applications. Additionally, the course should make familiar with existing web modelling approaches and enable for a critical reflection on their strength and applicability with respect to various internet application types.

Specific attention will be given to the model-driven development of "ubiquitous web applications" requiring for context-awareness.

*Contents:* Role of modelling in web engineering; Basics of model-driven engineering of internet applications; Details on specific approaches (WebML, OOWS, OO-H, UWE, etc.); Characteristics of "ubiquitous web applications"; Customization as mechanism to address context-awareness.

### **Product Line Engineering (2KV; SEA, Grünbacher)**

*Ziele:* Kennenlernen von Methoden des Product Line Engineering. Praktische Erprobung von Werkzeugen für das Domain Engineering und Application Engineering. Vertiefung anhand von Fallstudien.

*Inhalt:* Feature Modellierung, orthogonale Variabilitätsmodelle, Produktlinienarchitekturen, Architekturvariabilität und komponentenbasierte Software-Entwicklung. Prozessframeworks für Produktlinienentwicklung (PuLSE, KobrA), Ökonomische Aspekte von Produktlinien. PL Werkzeuge.

### **Rewriting in Logic and Computer Science (2VO; RISC, Winkler)**

*Goals and Content:* This lecture is intended to give an introduction to term rewriting systems and their application in computer science and logic. In particular we will discuss general reduction relations, properties of reduction systems such as confluence and termination, completion procedures (Knuth-Bendix procedure) for word rewriting systems and term rewriting systems.

Term rewriting systems developed out of mathematical logic and are an important part of theoretical computer science. They consist of sequences of discrete transformation steps where one term is replaced with another and have applications in many areas, from functional programming to automatic theorem proving and computer algebra.

### **Secure Code (1KV; FIM, NN)**

*Ziele:* Angriffe auf Anwendungen erfolgen nicht nur - wie häufig angenommen - durch unsicher konfigurierte Systeme. Immer häufiger werden Lösungen durch Angriffe auf Applikationsebene kompromittiert, z.B. durch das Absetzen von SQL-Anweisungen in einem fehlerhaft programmierten Webformular. Die Entwicklung sicherer Software ist daher eine Grundvoraussetzung für die Absicherung von unternehmenskritischen Anwendungen und muss dafür ein integraler Bestandteil des gesamten Entwicklungsprozesses sein - beginnend bei der Anwendungsarchitektur bis hin zu klaren Implementierungsstrategien.

*Inhalt:* Im Rahmen der Lehrveranstaltung wird zuerst an Hand von Beispielen gezeigt, welche dramatischen Auswirkungen tlw. einfache Programmierfehler auf die Sicherheit einer Applikation haben können. Anschließend werden Methoden und Technologien besprochen, die die Qualität der Applikation und damit auch Ihre Sicherheit verbessern.

### **Sensor Networks (2KV; PC, Externer Lehrbeauftragter, Koordination: Ferscha)**

*Ziele:* Sensornetze bestehen aus mehreren kleinen, autonomen und drahtlos kommunizierenden Geräten, die mit Sensoren ausgestattet sind und üblicherweise über begrenzte Ressourcen verfügen. Typische Einsatzgebiete sind die Beobachtung von Umweltaktivitäten (Wetter, Gletscher, Verhalten von Tieren in ihrem natürlichen Lebensraum) sowie die Überwachung von Bauwerken, Brücken, etc. In der LVA werden relevante Technologien sowie Konzepte für das Grundverständnis von Sensornetzen vorgestellt. In Einzel- bzw. Gruppenarbeiten wird anschließend das erarbeitete Wissen in der Praxis angewendet.

*Inhalt:* Anhand von Übungsbeispielen mit ZigBee-Technologie werden u.a. folgende Themen praktisch behandelt: Selbstorganisation von Knoten, Code-Verteilung (Over-the-air programming), Adressierung/Lokalisierung, Routing und Programmierabstraktionen, Single- und Multihop-Kommunikation, Energiesparmöglichkeiten, etc.

### **Sicherheit in Applikationsprotokollen (1KV; FIM, Dietmüller)**

*Ziele:* Die LVA gibt einen Überblick über eine Reihe verfügbarer Internet-Applikationsprotokolle. Anhand einfacher Programmierbeispiele wird das erworbene Wissen in die Praxis umgesetzt.

*Inhalt:* Protokolle, wie zum Beispiel HTTP, SMTP, POP3, IMAP, FTP, TELNET und deren Struktur und Aufbau. Ferner werden auch die möglichen Einsatzgebiete der Protokolle sowie deren Sicherheitsproblematiken erläutert.

### **Statistik 2 (2KV; CA, Pölz)**

*Ziele:* Vertiefung der in Statistik 1 vermittelten Inhalte.

*Inhalt:* Deskriptive Statistik, Datenanalyse mit SPSS, Analyse von Kontingenztafeln, Testtheorie (Aufstellen von Hypothesen, Fehler 1. und 2. Art, ...), Chi-Quadrat-Test, Chi-Quadrat-Maß und daraus abgeleitete Größen (Cramer's V, Phi, Kontingenzkoeffizient C), Fisher's Exact Test und Korrigiertes Chi-

Quadrat nach Yates, empirische Kovarianz, Korrelationskoeffizient nach Bravais-Pearson, Bestimmtheitsmaß, Regressionsanalyse, Kolmogoroff-Smirnov-Test, Einstichproben-t-Test, Zweistichproben-t-Test.

### **System Software (2KV; SSW, Mössenböck)**

*Ziele:* Studium wichtiger Systemprogramme wie Speicherverwaltung, Garbage Collectoren, Linker, Lader, Debugger und Texteditoren.

*Inhalt:* Speicherverwaltung (Mark/Free, Alloc/Dealloc, Buddy-System, Defragmentierung), Garbage Collection (Reference Counting, Mark & Sweep, Mark & Compact, Stop & Copy, Generation Scavenging, Inkrementelles GC, Konservatives GC), Linker und Lader, Debugger, Texte und Texteditoren.

### **Theoretical Concepts of Machine Learning (2VO+1UE; BIO, Bodenhofer)**

*Ziele:* Verständnis der theoretischen Grundlagen maschinellen Lernens, insbesondere der statistischen Lerntheorie; Vertiefung maschineller Lernverfahren (v.a. Support-Vektor-Maschinen und neuronale Netze) und ihrer mathematischen Grundlagen

*Inhalt:* kurze Wiederholung der wesentlichen Konzepte maschinellen Lernens; Evaluierungsmaße und Optimierungsstrategien; statistische Lerntheorie; Support-Vektor-Maschinen: Vertiefung und Anwendungen; neuronale Netze: Vertiefung und Anwendungen

### **Übersetzerbau 2 (2KV; SSW, Mössenböck)**

*Ziele:* Vertiefung im Gebiet des Übersetzerbaus, insbesondere in der Compiler-Optimierung und der Codeerzeugung.

*Inhalt:* Getrennte Übersetzung, Codeerzeugung für Registermaschinen (IA32), Zwischensprachen (abstrakter Syntaxbaum, Kontrollflussgraph, Dominatorbaum, Static Single Assignment Form), Optimierung (Common Subexpression Elimination, Inlining, Loop Unrolling, Loop-Invariant Code Motion, ...), Registerallokation.

### **VLSI-Entwurf (2KV; RIIC, Spilka)**

*Ziele:* Vertiefen der Kenntnisse im Bereich Entwurf komplexer digitaler Schaltungen mittels VHDL. Überblick über einen Designflow in der Zieltechnologie ASIC-Standardzellen.

*Inhalt:* Grundlagen für einen Entwurf und Synthese eines 16-Bit-RISC-Prozessorkerns als Full-Costum-IC, aufbauend auf bestehenden VHDL-Kenntnissen. Simulation des Gesamtsystems mit Timingdaten. Kennen lernen einer Synthese-Bibliothek und vom Synthese-Tool „DesignVision“.

### **Web Engineering (2KV; FAW, Pröll)**

*Ziele:* Grundlagen des Web-Engineering und seiner diversen Aspekte.

*Inhalt:* Aufgaben und Phasen des Web-Engineering, WebIS-Architekturen, Web&DB, Beispiel Tiscover, Spezielle Themen des Web Engineering (Modellierung, Usability, Performanz/Stabilität, Caching, Testen, Statistiken, Security), Behandlung eines aktuellen Schwerpunktthemas (z.B. Personalisierung)

### **Web Information Retrieval (2KV; FAW, Pröll)**

*Ziele:* Grundlagen und Techniken des Web Information Retrieval und deren Einsatz in Suchmaschinen.

*Inhalt:* Information Retrieval "in a nutshell", Konzepte des Web Information Retrieval (Web-Crawler etc.), Search Engines und Web Site Search (Lucene et al.), Search Engine Optimization, Web Information Extraction

### **Wireless LANs (1KV; FIM, Praher)**

*Ziele:* Einführung in Technologie, Standards, Aufbau und Einsatz von Wireless LANs.

*Inhalt:* Einleitung: Gründe für und gegen WLANs, Komponenten eines WLANs: Basiskomponenten, Basic Service Set, Extended Service Set, Distributionssystem; der Standard 802.11 und 802.11b: CSMA/CA, DCF, PCF, Beacon, FHSS, DSSS, Physical Layers, Frame Formats etc., Verbindungsaufbau, Roaming; Sicherheit in WLANs: WEP, WPA, 802.1x (EAP) 802.11i; weitere Varianten und Entwicklungen des Standards: 802.11a, 802.11h, 802.11g, 802.11n, 802.11f (Roaming), 802.11e (QoS); Meshed Networks: Wireless Distribution auf IP-Ebene, proaktives und reaktives Routing, hybride Routingverfahren; Planung eines WLANs: Sicherheit, Mobilität, Leistungsanforderungen, Netzwerkgestaltung, Tuning; WLAN und andere drahtlose Kommunikationstechniken: WLAN und Bluetooth, WLAN und UMTS; Das WLAN an der JKU; Das LAN der Zukunft.