

Modulkatalog

Master-Studiengang Angewandte Informatik Prüfungs- und Studienordnung 2015

***Fachbereich Information und Kommunikation
Hochschule Flensburg***

Stand: März 2016

Es folgen die Modulbeschreibungen der Wahlpflichtmodule, die im 1. oder 2. Semester des Master-Studiums Angewandte Informatik belegt werden können.

Jedes Wahlpflichtmodul ist mit 6 cp bewertet. Insgesamt müssen Wahlpflichtmodule im Umfang von mindestens 36 cp belegt werden. Die Wahlpflichtmodule sind zum Teil den Schwerpunkten Internet-Sicherheit, Mobile Computing und Human-Computer Interaction zugeordnet.

Dieser Modulkatalog wird semesterweise aktualisiert. Die meisten Wahlpflichtmodule werden in regelmäßigem Turnus angeboten, manche dagegen unregelmäßig.

Wenn mehrere Prüfungsformen angegeben sind, bestimmt der oder die Modulverantwortliche zu Beginn des Semesters die Prüfungsform, die in dem betreffenden Semester gilt, und kündigt diese den Studierenden an.

Abkürzungen für Prüfungsformen sind:

K(n)	Klausur (Dauer in Stunden)
HA	Hausaufgaben
Arb	Schriftliche Ausarbeitung
Votr	Vortrag

Hinweis:

Da sich die Modulbeschreibungen in erster Linie an Studierende bzw. Studieninteressierte richten, sind die Kompetenzen in den Lernzielen in persönlicher Ansprache mit "Sie" formuliert, etwa: "Darüber hinaus sind Sie in der Lage, ..."

Inhaltsverzeichnis

Big Data - Darstellung und Analyse großer Datenmengen	4
Buildmanagement und Testautomatisierung mobiler Anwendungen	5
Diskrete Optimierung und Operations Research	6
High Performance Computing	7
Hot Topics in IT Security	8
Human-Computer Interaction	9
Informationsvisualisierung	10
Kryptologie und Systemsicherheit	11
Leistungsanalyse von Netzen	12
Medizinische Visualisierung	13
Mobile Engineering	14
Mobile Webentwicklung	15
Prinzipien von Programmiersprachen	16
Quanten-Computing	17
Reaktive Programmierung	18
Sicherheit von mobilen Applikationen	19
Sicherheitsmanagement in Netzen	20
Usability Engineering	21

Big Data - Darstellung und Analyse großer Datenmengen

Modulbezeichnung:	Big Data - Darstellung und Analyse großer Datenmengen
Modulverantwortliche:	Prof. Dr. M. Zachow
Lehrende:	Prof. Dr. M. Zachow
Turnus:	unregelmäßig
Sprache:	deutsch
Schwerpunkt:	Human-Computer Interaction
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 120 h, Gesamtaufwand: 180 h
Leistungspunkte:	6 cp
Lernziele / Kompetenzen:	Sie verstehen die Grundlagen von Big Data, sowohl technologisch als auch methodisch. Sie können Big-Data-Projekte planen, beurteilen, umsetzen und betreiben. Sie kennen Verfahren zur Aufbereitung, Analyse und Darstellung von Daten und können diese kontextgebunden einsetzen.
Inhalt:	<p>Vorlesung</p> <ul style="list-style-type: none"> • Grundlagen und Einführung • Datenbank-Technologien aus dem NoSQL-Umfeld • Ecosysteme und Infrastrukturen (z.B. Apache Hadoop) • Verteilte Datenbanken, Clustering und Cloud-Lösungen • Business Intelligence • Analyse, Algorithmen und Machine Learning • Datenvisualisierung <p>Labor</p> <p>In den begleitenden Laboren wird die Datenanalyse und Visualisierung existierender Datensets (Stichwort Open Data) praktisch erprobt.</p>
Prüfungsform:	HA, Arb, Votr
Literatur:	D. MINER, A.SHOOK: MapReduce Design Patterns. O'Reilly (2012) R. WARTALA: Hadoop. Open Source Press (2012)

Buildmanagement und Testautomatisierung mobiler Anwendungen

Modulbezeichnung:	Buildmanagement und Testautomatisierung mobiler Anwendungen
Modulverantwortliche:	Prof. Dr. M. Zachow
Lehrende:	Prof. Dr. M. Zachow
Turnus:	im Wintersemester
Sprache:	deutsch
Schwerpunkte:	Mobile Computing, Human-Computer Interaction
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 120 h, Gesamtaufwand: 180 h
Leistungspunkte:	6 cp
Lernziele / Kompetenzen:	Ihnen ist bewusst, dass Qualitätssicherung für mobile Geräte wegen der Fragmentierung und Hardware-Abhängigkeit ein aufwendiger Prozess ist. Sie kennen Ansätze und Technologien, die helfen, Tests für mobile Anwendungen zu automatisieren und mit unterschiedlichster Hardware durchzuführen. Sie überblicken die Vertriebs- und Auslieferungswege mobiler Anwendungen und kennen die Prozesse, die nötig sind, um Anwendungen auf verschiedensten mobilen Geräten und Betriebssystemen auszuliefern.
Inhalt:	<p>Vorlesung</p> <ul style="list-style-type: none"> • Grundlagen und Einführung • Buildprozesse für mobile Applikationen, Signing • Vertriebswege für Apps • Verteilung mobiler Applikationen im Test-Szenario • Testing Frameworks für mobile Anwendungen • Continuous Integration für mobile Anwendungen • Testing über cloudbasierte Dienste <p>Labor</p> <p>In den begleitenden Laboren üben Sie anhand von Beispiel-Anwendungen, den Build- und Testprozess für verschiedene Betriebssysteme zu automatisieren.</p>
Prüfungsform:	HA, Arb, Votr
Literatur:	J. BAUMANN: Gradle. dpunkt (2013) H.M. SNEED, M. BAUMGARTNER, R. SEIDL: Der Systemtest. 3. Auflage, Hanser (2011)

Diskrete Optimierung und Operations Research

Modulbezeichnung:	Diskrete Optimierung und Operations Research
Modulverantwortlicher:	Prof. Dr. M. Krätzschar
Lehrender:	Prof. Dr. M. Krätzschar
Turnus:	im Wintersemester
Sprache:	deutsch
Schwerpunkt:	--
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 120 h, Gesamtaufwand: 180 h
Leistungspunkte:	6 cp
Lernziele / Kompetenzen:	Sie verstehen die theoretischen Grundlagen und Ansätze der mathematischen Optimierung. Sie können lineare und insbesondere auch diskrete Optimierungsprobleme aufstellen und in AMPL (A Mathematical Programming Language) beschreiben. Ferner kennen Sie Verfahren zur numerischen Lösung von Optimierungsproblemen und können entsprechende Programme anwenden.
Inhalt:	<p>Vorlesung</p> <ul style="list-style-type: none"> • Lineare Optimierungsprobleme • Simplex-Algorithmus • Einführung in AMPL • Ganzzahlige Optimierungsprobleme • Assignment-Probleme • Binäre Optimierungsprobleme • Reihenfolge-Probleme <p>Labor</p> <p>In den begleitenden Übungen arbeiten Sie mit AMPL und setzen Programme zur numerischen Behandlung von Optimierungsproblemen ein.</p>
Prüfungsform:	K(2)
Literatur:	R. FOURER, D. GAY, B. KERNIGHAN: AMPL - A Modeling Language for Mathematical Programming. Duxbury (1997) P. GRITZMANN: Grundlagen der Mathematischen Optimierung. 2. Auflage, Springer Spektrum (2013) A. KOOP, H. MOOCK: Lineare Optimierung. 2. Auflage, Spektrum (2008)

High Performance Computing

Modulbezeichnung:	High Performance Computing
Modulverantwortlicher:	Prof. Dr. H.W. Lang
Lehrender:	Prof. Dr. H.W. Lang
Turnus:	unregelmäßig
Sprache:	deutsch
Schwerpunkt:	--
Lehrform / SWS:	Workshop, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 120 h, Gesamtaufwand: 180 h
Leistungspunkte:	6 cp
Lernziele / Kompetenzen:	Sie kennen die unterschiedlichen Konzepte paralleler Computer-Architekturen und paralleler Algorithmen, ferner können Sie parallele Algorithmen hinsichtlich ihrer Komplexität analysieren und bewerten. Sie können GPU-Programme in OpenCL bzw. CUDA implementieren, und Sie haben Erfahrung bei der Umsetzung entsprechender Problemlösungen mittels GPU-Computing.
Inhalt:	<p>Vorlesung</p> <ul style="list-style-type: none"> • Parallele Computer-Architekturen, Prinzipien der Parallelverarbeitung: SIMD, MIMD, Systolische Felder • Parallele Algorithmen: Entwurfsprinzipien, Kommunikationskomplexität • GPU-Computing: OpenCL, CUDA • Anwendungen und Implementierung von GPU-Programmen: Zahlentheorie und Kryptografie, DNA-Sequenzvergleich, Neuronale Netze <p>Labor</p> <p>Im Labor werden parallele Algorithmen auf einem massiv parallelen GPU-System (nVidia K6000 bzw. Tesla K40c) implementiert und hinsichtlich der erzielten Leistung bewertet.</p>
Prüfungsform:	HA, Arb, Votr
Literatur:	P. PACHECO: Parallel Programming. Morgan Kaufmann (2011) N. WILT: The CUDA Handbook. Addison Wesley (2013)

Hot Topics in IT Security

Modulbezeichnung:	Hot Topics in IT Security
Modulverantwortlicher:	Prof. Dr. S. Gajek
Lehrender:	Prof. Dr. S. Gajek
Turnus:	im Sommersemester
Sprache:	deutsch
Schwerpunkt:	Internet-Sicherheit
Lehrform / SWS:	Workshop, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 120 h, Gesamtaufwand: 180 h
Leistungspunkte:	6 cp
Lernziele / Kompetenzen:	Sie besitzen ein breites Wissen über noch ungelöste Probleme der IT-Sicherheit in Industrie und Wissenschaft (<i>Hot Topics</i>). Sie verstehen moderne Verfahren der Kryptografie und IT-Sicherheit. Sie sind in der Lage, entsprechende kryptografische Protokolle zur Lösung von aktuellen Sicherheitsproblemen (z.B. im Cloud Computing oder Internet-of-Things) einzusetzen und hinsichtlich ihrer Sicherheitsgarantien zu evaluieren.
Inhalt:	<p>Es werden verschiedene Themen der modernen, praxisorientierten Kryptografie und IT-Sicherheit behandelt. Unter anderem werden folgende Verfahren vorgestellt:</p> <ul style="list-style-type: none"> • Key Exchange and Distribution Protokolle und deren Anwendung in sicheren Kommunikationsprotokollen (SSL/TLS, IPSec) • Multi-Party Computation Protokolle für die private Analyse von großen Datenmengen, wie sie für Data-Mining-Anwendungen erforderlich sind • Kurze, nicht-interaktive Beweissysteme und deren Anwendung für die Verifikation von Cloud Computing • Funktionale Verschlüsselung, um Cloud Storage Systeme mit einer Zugangspolicy zu verschlüsseln <p>Zum Abschluss werden die erlernten Inhalte in konkreten Beispielen angewendet, um zu neuen oder verbesserten Sicherheitslösungen zu gelangen.</p>
Prüfungsform:	HA, Arb, Votr
Literatur:	<p>O. GOLDRICH: Foundations of Cryptography Vol. 1. Cambridge University Press (2008)</p> <p>O. GOLDRICH: Foundations of Cryptography Vol. 2. Cambridge University Press (2009)</p> <p>J. KATZ, Y. LINDELL: Introduction to Modern Cryptography. 2. Auflage, Taylor & Francis (2014)</p>

Human-Computer Interaction

Modulbezeichnung:	Human-Computer Interaction
Modulverantwortlicher:	Prof. Dr. M. Teistler
Lehrende:	Prof. Dr. M. Teistler, S. Reinhold, M.A., V. Schomakers, B.A.
Turnus:	im Sommersemester
Sprache:	deutsch
Schwerpunkt:	Human-Computer Interaction
Lehrform / SWS:	, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 120 h, Gesamtaufwand: 180 h
Leistungspunkte:	6 cp
Lernziele / Kompetenzen:	
Inhalt:	<p>In kleinen Gruppen werden ausgewählte Themen im Bereich Human-Computer Interaction bearbeitet.</p> <p>Ziel der Veranstaltung ist, ein wissenschaftliches Paper oder Poster anzufertigen und bei einer einschlägigen Konferenz einzureichen, z.B. bei</p> <ul style="list-style-type: none"> • MuC - Mensch und Computer http://muc2016.mensch-und-computer.de/cfp/mensch-computer-interaktion/ • The ACM Symposium on Virtual Reality Software and Technology (VRST) http://www.vrst2016.lrz.de/call-for-participation/ • NordiCHI - Game-Changing Design http://www.nordichi2016.org/participate/posters/
Prüfungsform:	
Literatur:	A. BUTZ, A. KRÜGER: Mensch-Maschine-Interaktion. De Gruyter Oldenbourg (2014)

Informationsvisualisierung

Modulbezeichnung:	Informationsvisualisierung
Modulverantwortlicher:	Prof. Dr. K. Hartmann
Lehrender:	Prof. Dr. K. Hartmann
Turnus:	im Sommersemester
Sprache:	deutsch
Schwerpunkt:	Human-Computer Interaction
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 120 h, Gesamtaufwand: 180 h
Leistungspunkte:	6 cp
Lernziele / Kompetenzen:	Sie können Diagramme lesen und konstruieren. Sie sind in der Lage, Evaluationen zu entwerfen und durchzuführen, um die Effektivität von Visualisierungen gezielt zu untersuchen. Sie können eigene Visualisierungen mit Hilfe üblicher Programmiersprachen umsetzen.
Inhalt:	<p>Das Ziel des relativ neuen Forschungszweiges Informationsvisualisierung ist es, die Vorzüge des visuellen Wahrnehmungskanals zur parallelen Verarbeitung einer Vielzahl von Informationen zu nutzen. Durch Visualisierung wird die Prozesskette Sehen - Erkennen - Verstehen aktiviert. So können große Datenmengen interaktiv erkundet und Hypothesen überprüft werden.</p> <p>Vorlesung</p> <ul style="list-style-type: none"> • Visualisierungsziele, Visualisierungspipeline • Datenbeschreibung und -auswahl • Mapping: von Daten zur Geometrie • Techniken der Datenvisualisierung • Kartierung abstrakter Daten • Graphen-Layout • Annotationen in Visualisierungen <p>Labor</p> <p>In den begleitenden Laborübungen erproben Sie praktische die Datenanalyse und die Visualisierung vorhandener realer Datensätze.</p>
Prüfungsform:	HA, Arb, Votr, K(2)
Literatur:	<p>B.B. BEDERSON, B. SHNEIDERMAN (HRSG.): The Craft of Information Visualization. Morgan Kaufmann (2003)</p> <p>M. CARD, S. CARD: Readings in Information Visualisation. Using Vision to Think. Morgan Kaufmann (1999)</p> <p>H. SCHUMANN, W. MÜLLER: Visualisierung. Springer (1999)</p> <p>C. WARE: Information Visualization: Perception for Design. 3. Auflage, Morgan Kaufmann (2012)</p>

Kryptologie und Systemsicherheit

Modulbezeichnung:	Kryptologie und Systemsicherheit
Modulverantwortlicher:	Prof. Dr. M. Kersken
Lehrender:	Prof. Dr. M. Kersken
Turnus:	im Wintersemester
Sprache:	deutsch
Schwerpunkt:	Internet-Sicherheit
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 120 h, Gesamtaufwand: 180 h
Leistungspunkte:	6 cp
Lernziele / Kompetenzen:	Sie kennen die mathematischen Grundlagen verschiedener kryptografischer Verfahren und zugehöriger Angriffs-Szenarien. Sie können die Sicherheit dieser Verfahren sowohl theoretisch als auch aufgrund Ihrer praktischen Erfahrungen bewerten.
Inhalt:	<p>Kryptografie ist heute mehr als nur das Verschlüsseln von geheimen Botschaften - kryptografische Methoden werden verwendet, um Dokumente zu signieren, um Kommunikationspartner zu authentifizieren, um Teilnehmer zu anonymisieren und vieles mehr. Dies alles sind unterschiedliche Aspekte der Systemsicherheit. Die tatsächliche Systemsicherheit hängt dabei von der Stärke der kryptografischen Verfahren ab. Die Stärke ausgewählter kryptografischer Verfahren zu untersuchen ist Gegenstand dieser Lehrveranstaltung.</p> <p>Vorlesung</p> <ul style="list-style-type: none"> • Kollisionssicherheit von Hash-Verfahren • Faktorisierung großer Zahlen • Elliptische-Kurven-Kryptografie <p>Labor</p> <p>Im Labor werden verschiedene kryptografische Verfahren praktisch untersucht und im Hinblick auf den Rechenaufwand bewertet, der zum Brechen der Verfahren erforderlich ist.</p>
Prüfungsform:	K(2)
Literatur:	<p>R. KÜSTERS, T. WILKE: Moderne Kryptographie. Vieweg+Teubner (2011)</p> <p>A. WERNER: Elliptische Kurven in der Kryptographie. Springer (2013)</p> <p>S.Y. YAN: Computational Number Theory and Modern Cryptography. Wiley (2013)</p>

Leistungsanalyse von Netzen

Modulbezeichnung:	Leistungsanalyse von Netzen
Modulverantwortlicher:	Prof. Dr. T. Uhl
Lehrender:	Prof. Dr. T. Uhl
Turnus:	unregelmäßig
Sprache:	deutsch
Schwerpunkt:	--
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 120 h, Gesamtaufwand: 180 h
Leistungspunkte:	6 cp
Lernziele / Kompetenzen:	Sie kennen die Methoden für die Leistungsanalyse von Warteschlangennetzen und können diese in der Praxis anwenden.
Inhalt:	<p>Vorlesung</p> <ul style="list-style-type: none"> • Klassifizierung von Systemen und Modellen • Zielsetzung beim Systementwurf • Wahrscheinlichkeitsrechnung, mathematische Statistik, Warteschlangentheorie • Exakte Methoden zur Leistungsanalyse von Warteschlangenmodellen (Theorem von Jackson, Theorem von Gordon/Nowell, BCMP Theorem) • Numerische Methoden zur Leistungsanalyse (Anwendung der Markoff-Ketten) • Klassifizierung der Simulationsmethoden • Zufallszahlen: Erzeugung, Transformation • Simulation von Kommunikationssystemen, Simulationsprachen <p>Labor</p> <p>In begleitenden Übungen werden die erlernten Konzepte praktisch angewendet und im Computerlabor umgesetzt.</p>
Prüfungsform:	HA, Arb, Votr, K(2)
Literatur:	<p>K. BAUKNECHT J. KOHLAS: Simulationstechnik. Springer (1976)</p> <p>G. BOLCH: Leistungsbewertung von Rechensystemen. Springer (1989)</p> <p>L. KLEINROCK: Queueing Systems, Vol. 1. Wiley (1976)</p> <p>A. PAPOULIS: Probability, Random Variables and Stochastic Processes. 4. Auflage, McGraw-Hill (2002)</p>

Medizinische Visualisierung

Modulbezeichnung:	Medizinische Visualisierung
Modulverantwortlicher:	Prof. Dr. M. Teistler
Lehrender:	Prof. Dr. M. Teistler
Turnus:	im Wintersemester
Sprache:	deutsch
Schwerpunkte:	Mobile Computing, Human-Computer Interaction
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 120 h, Gesamtaufwand: 180 h
Leistungspunkte:	6 cp
Lernziele / Kompetenzen:	Sie kennen die medizinischen bildgebenden Verfahren und deren klinische Bedeutung. Sie sind mit unterschiedlichen Techniken zur Visualisierung medizinischer Bilddaten vertraut und Sie sind in der Lage, Lösungen aus dem Bereich Human-Computer Interaction zu erarbeiten, um die Benutzbarkeit radiologischer Software zu verbessern.
Inhalt:	<p>Vorlesung</p> <ul style="list-style-type: none"> • Einführung in die verschiedenen Verfahren der medizinischen Bildgebung (Röntgen, Sonographie, Computertomographie, Kernspintomographie u.a.) • Klinische Bedeutung der medizinischen Bildgebung / wie arbeitet ein Radiologe? • Visualisierungstechniken • Medizinische Visualisierung und Informatik: ausgewählte Themen <p>Hausarbeit</p> <p>Entwicklung und Präsentation von Konzepten und Softwareprototypen als Gruppenarbeit; hier sind verschiedene Themen möglich, auch eigene Vorschläge insbesondere aus den Bereichen Human-Computer-Interaction, Usability, 3D-Visualisierung.</p> <p>Beispielprojekte aus dem Themenumfeld: http://www.userad.info</p> <p>Je nach Schwerpunkt der Hausarbeit kann dieses Modul den Schwerpunkten [Hci] oder [Mob] zugeordnet werden.</p>
Prüfungsform:	HA, Arb, Votr
Literatur:	O. DÖSSEL: Bildgebende Verfahren in der Medizin. 2. Auflage, Springer Vieweg (2016) H. HANDELS: Medizinische Bildverarbeitung. 2. Auflage, Vieweg+Teubner (2009)

Mobile Engineering

Modulbezeichnung:	Mobile Engineering
Modulverantwortlicher:	Prof. Dr. T. Aschmoneit
Lehrender:	Prof. Dr. T. Aschmoneit
Turnus:	im Sommersemester
Sprache:	deutsch
Schwerpunkt:	Mobile Computing
Lehrform / SWS:	Workshop, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 120 h, Gesamtaufwand: 180 h
Leistungspunkte:	6 cp
Lernziele / Kompetenzen:	Sie kennen die Architekturen unterschiedlicher markt-relevanter Plattformen für Mobile Devices (iOS, Android, ...) und deren Derivate z.B. für Wearables, TV, Automotive. Sie kennen die Probleme, die bei der Entwicklung von Apps für diese unterschiedlichen Plattformen auftreten und können Lösungen entwickeln, die auf dem Einsatz von Tools oder auf der Entwicklung portabler Softwarekomponenten basieren.
Inhalt:	<p>Themen</p> <p>Architektur mobiler Plattformen</p> <ul style="list-style-type: none"> • iOS, Android, ... <p>Cross-Platform Development (CPD)</p> <ul style="list-style-type: none"> • Problembereiche • Prinzipielle Lösungsstrategien des CPD (UI, Echtzeit, Storage, Security) • Tools • Framework für CPD <p>App Entwicklung</p> <ul style="list-style-type: none"> • User Interface • Wearables • Battery Management • Mobile spezifische Cloud Services • Streaming: Voice und Video <p>Im praktischen Teil entwickeln Sie Apps unter Verwendung der behandelten Cross-Platform Development-Strategien.</p>
Prüfungsform:	HA, Arb, Votr, K(2)
Literatur:	D. HERMES: Xamarin Mobile Application Development. Apress (2016) G. SHACKLES: Mobile Development with C#. O'Reilly (2012)

Mobile Webentwicklung

Modulbezeichnung:	Mobile Webentwicklung
Modulverantwortliche:	Prof. Dr. M. Zachow
Lehrende:	Prof. Dr. M. Zachow
Turnus:	im Sommersemester
Sprache:	deutsch
Schwerpunkt:	Mobile Computing
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 120 h, Gesamtaufwand: 180 h
Leistungspunkte:	6 cp
Lernziele / Kompetenzen:	Sie können Web-Anwendungen entwerfen und umsetzen, die auf verschiedensten Endgeräten funktionieren. Sie kennen aktuelle Konzepte und Tools, die bei der Entwicklung solcher Anwendungen unterstützen. Sie sind mit hybriden Applikationen vertraut.
Inhalt:	<p>Vorlesung</p> <ul style="list-style-type: none"> • Grundlagen und Einführung • Zugriff auf System-Funktionen wie Kamera • Offline-Applikationen • Konzepte (wie momentan z.B. Mobile First oder Responsive Webdesign) • Aktuelle Libraries • Entwicklungs- und Test-Tools • Hybride Apps • Monetarisierung <p>Labor</p> <p>In den begleitenden Laboren wird eine Web-Anwendung für mobile Geräte als individuelles Projekt umgesetzt.</p>
Prüfungsform:	HA, Arb, Votr
Literatur:	A. ERTEL, K. LABORENZ: Responsive Webdesign. Galileo Computing (2013) J. NIELSEN, R. BUDIU: Mobile Usability. 2. Auflage, mitp (2013)

Prinzipien von Programmiersprachen

Modulbezeichnung:	Prinzipien von Programmiersprachen
Modulverantwortlicher:	Prof. Dr. H.W. Lang
Lehrende:	Prof. Dr. H.W. Lang, Dr. P. Sadeghi
Turnus:	unregelmäßig
Sprache:	deutsch
Schwerpunkt:	--
Lehrform / SWS:	Vorlesung / Übungen / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 120 h, Gesamtaufwand: 180 h
Leistungspunkte:	6 cp
Lernziele / Kompetenzen:	<p>Sie beherrschen die grundlegenden Techniken der formalen Verifikation von Programmen. Sie können die Korrektheit von kleineren Programmstücken formal beweisen.</p> <p>Ferner kennen Sie die grundlegenden Konzepte der funktionalen Programmierung. Sie können funktionale Programme in der Programmiersprache Haskell schreiben.</p>
Inhalt:	<p>Thematisch geht es unter anderem um unterschiedliche Aspekte der Korrektheit von Programmen.</p> <p>Programmverifikation (Lang)</p> <p><i>Testing can only show the presence of errors, not their absence</i> (Dijkstra). Dagegen ist es mit den Methoden der Programmverifikation möglich, formal zu beweisen, dass ein Programm fehlerfrei ist (d.h. die Spezifikation erfüllt).</p> <ol style="list-style-type: none"> 1. Aussagenlogik 2. Korrektheitsformeln, Wertzuweisungsaxiom 3. Beweis von If-Anweisungen 4. Korrektheit von While-Schleifen 5. Terminierung von Schleifen <p>Funktionale Programmierung (Sadeghi)</p> <p>In der funktionalen Programmierung wird spezifiziert, <i>was</i> berechnet werden soll, nicht <i>wie</i> es berechnet werden soll. Spezifikation und Programm sind im wesentlichen identisch, daher ist Korrektheit automatisch gegeben.</p> <ol style="list-style-type: none"> 1. Programmiersprache Haskell 2. Rekursion und Iteration 3. Lambda-Ausdrücke 4. Funktionen höherer Ordnung 5. Lazy evaluation <p>In begleitenden Übungen werden die erlernten Konzepte praktisch angewendet und geübt und soweit möglich im Computerlabor umgesetzt. Hierbei werden Python und Haskell als Programmiersprachen eingesetzt.</p>
Prüfungsform:	Sonstige Prüfungsleistung
Literatur:	<p>G. HUTTON: Programming in Haskell. Cambridge University Press (2007)</p> <p>H.W. LANG: Algorithmen in Java. 3. Auflage, Oldenbourg (2012)</p> <p>P. PEPPER, P. HOFSTEDT: Funktionale Programmierung.. 2. Auflage, Springer (2006)</p>

Quanten-Computing

Modulbezeichnung:	Quanten-Computing
Modulverantwortlicher:	Prof. Dr. M. Krätzschar
Lehrender:	Prof. Dr. M. Krätzschar
Turnus:	unregelmäßig
Sprache:	deutsch
Schwerpunkt:	--
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 120 h, Gesamtaufwand: 180 h
Leistungspunkte:	6 cp
Lernziele / Kompetenzen:	Sie verstehen die theoretischen Grundlagen von Quanten-Computing und kennen die Möglichkeiten, die sich durch den Einsatz von Quanten-Algorithmen in Zukunft (möglicherweise) ergeben werden.
Inhalt:	<p>Vorlesung</p> <ul style="list-style-type: none"> • Quanten, Quantenphysik und Quantenmechanik • Mathematische Grundlagen der Quantenphysik • Qubits, Quantenregister, Verschränkungen • Quantenschaltungen • Quanten-Fourier-Transformation (QFT) • QFT-Algorithmen • Quanten-Suchalgorithmen • Quantenrechner
Prüfungsform:	K(2)
Literatur:	P. KAYE, R. LAFLAMME, M. MOSCA: An Introduction to Quantum Computing. Oxford University Press (2007) A. DE VRIES: Quantenrechnen. Books on Demand (2012)

Reaktive Programmierung

Modulbezeichnung:	Reaktive Programmierung
Modulverantwortlicher:	Prof. Dr. J. Christiansen
Lehrender:	Prof. Dr. J. Christiansen
Turnus:	im Sommersemester
Sprache:	deutsch
Schwerpunkt:	Mobile Computing
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 120 h, Gesamtaufwand: 180 h
Leistungspunkte:	6 cp
Lernziele / Kompetenzen:	Sie kennen die Grundprinzipien der reaktiven Programmierung und sind mit den wichtigsten Kombinatoren vertraut. Sie können selbstständig einfache Probleme mit Hilfe von reaktiver Programmierung in einer beliebigen Programmiersprachen lösen und sind in der Lage, das Verhalten anhand sogenannter "Marble Diagrams" zu visualisieren.
Inhalt:	<p>Die Ideen der reaktiven Programmierung erlauben es, die Veränderungen von Benutzeroberflächen bei der Änderung der zugrundeliegenden Modelle elegant zu beschreiben. Beim Einsatz von konventionellen Techniken wird die Implementierung eines solchen reaktiven Verhaltens schnell unübersichtlich.</p> <p>Neben der Anwendung im Bereich von Benutzeroberflächen wird die Implementierung von asynchronen Berechnungen mit Hilfe der reaktiver Programmierung demonstriert und der Zusammenhang mit anderen Konzepten zur Strukturierung von Programmen wie Promises erläutert.</p> <p>Die Vorlesung stellt die Prinzipien der reaktiven Programmierung und ihren Einsatz zur Implementierung von mobilen Anwendungen und von Web-Anwendungen vor.</p> <p>Zuerst werden die Grundlagen der funktionalen Programmierung eingeführt und es wird gezeigt, wie diese bei der reaktiven Programmierung zum Einsatz kommen. Für praktische Beispiele und Programmierübungen werden die sogenannten Reactive Extensions (Rx) für Java und JavaScript eingesetzt. Die Reactive Extensions wurden ursprünglich für die .NET-Plattform entwickelt, inzwischen stehen entsprechende Bibliotheken für eine Vielzahl von Programmiersprachen, unter anderem Java, Objective-C, Ruby, JavaScript, Scala und Swift zur Verfügung.</p>
Prüfungsform:	HA, Arb, Votr, K(2)
Literatur:	

Sicherheit von mobilen Applikationen

Modulbezeichnung:	Sicherheit von mobilen Applikationen
Modulverantwortlicher:	Prof. Dr. T. Aschmoneit
Lehrende:	Prof. Dr. T. Aschmoneit, A. Irawan, M.Sc.
Turnus:	im Wintersemester
Sprache:	deutsch
Schwerpunkte:	Internet-Sicherheit, Mobile Computing
Lehrform / SWS:	Workshop, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 120 h, Gesamtaufwand: 180 h
Leistungspunkte:	6 cp
Lernziele / Kompetenzen:	<p>Sie kennen die Angriffsszenarien auf mobile Systeme und können Schutzziele für mobile Systeme formulieren, Sicherheitsanalysen durchführen und wirksame Schutzmaßnahmen entwickeln.</p> <p>Im Detail kennen Sie mobile Plattformen auf Basis von iOS und Android sowie die dort verwendeten Kommunikationssysteme (LTE, Bluetooth, NFC, WLAN, ...). Im Besonderen sind Sie mit den Sicherheitsmechanismen der Plattformen (Execution Environments, SandBox, Signatur, Permissions, ...) und der Kommunikationswege (Authentication, Key Management und -Exchange, Verschlüsselung, ...) vertraut.</p>
Inhalt:	<p>Themen</p> <p>Grundlagen</p> <ul style="list-style-type: none"> • Übersicht mobile Plattformen • Mobile Kommunikation <p>Sicherheitsmechanismen</p> <ul style="list-style-type: none"> • Plattform • Betriebssystem • Web-Browser • Applikationen • Kommunikation (Luftschnittstelle, Netzwerk-Architektur) <p>Gefährdungen</p> <p>Schutzmaßnahmen</p> <p>Im Workshop halten Sie Vorträge zu den obigen Themen und führen praktische Versuche in folgenden Bereichen durch: Rooting, Forensic-Analysis (mit Einführung in die entsprechenden Tools), Analyse App-Kommunikation, Spyware- und Adware-Apps, APK-Decompile, NFC-Security und Eavesdropping Bluetooth.</p>
Prüfungsform:	HA, Arb, Votr, K(2)
Literatur:	<p>A. HOGG: Android Forensics. Syngress (2012)</p> <p>K. MAKAN, S. ALEXANDER-BROWN: Android Security Cookbook. Packt (2013)</p> <p>J. ZDZIARSKI: Hacking and Securing iOS Applications. O'Reilly (2012)</p>

Sicherheitsmanagement in Netzen

Modulbezeichnung:	Sicherheitsmanagement in Netzen
Modulverantwortlicher:	N.N.
Lehrender:	N.N.
Turnus:	unregelmäßig
Sprache:	deutsch
Schwerpunkt:	Internet-Sicherheit
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 120 h, Gesamtaufwand: 180 h
Leistungspunkte:	6 cp
Lernziele / Kompetenzen:	Sie kennen die unterschiedlichen Angriffsmöglichkeiten auf die Integrität von Computernetzen. Sie sind in der Lage, entsprechende Vorsorge- und Abwehrmaßnahmen hinsichtlich ihrer Wirksamkeit zu beurteilen und praktisch zu implementieren.
Inhalt:	<p>Vorlesung</p> <ul style="list-style-type: none"> • Begriffe und Definitionen zum Thema Netzwerksicherheit • Formen und Methoden von Angriffen • Angriffe auf Software (Pufferüberlauf, Stack Smashing, Formatstring) • Angriffe auf Netzwerkprotokolle (Man-in-the-Middle, Tunneling) • Angriffe auf die Netzstruktur (DoS: Denial of Service) • Tarnung von Angriffen (Fragmentierung, Spoofing) • Verwandte Angriffe (Social Engineering, Würmer, Trojanische Pferde, Dialer, Viren, Phishing) • Vorsorge (Authentifizierung, Autorisierung, Zertifizierung) • Sicherheitsinfrastruktur: Protokolle, Architekturen und Komponenten (Kerberos, X.509, IPsec, SSL/TLS, EAP, Firewalls, IDS, Honeypots) <p>Labor</p> <p>In begleitenden Übungen werden die erlernten Konzepte praktisch angewendet und im Computerlabor umgesetzt.</p>
Prüfungsform:	K(2)
Literatur:	G. SCHÄFER, M. ROßBERG: Netzsicherheit. 2. Auflage, dpunkt (2014) C. SORGE, N. GRUSCHKA, L. LO IACONO: Sicherheit in Kommunikationsnetzen. Oldenbourg (2013) M. SCHUMACHER, U. ROEDIG, M.L. MOSCHGATH: Hacker Contest. Springer (2002)

Usability Engineering

Modulbezeichnung:	Usability Engineering
Modulverantwortlicher:	Prof. Dr. M. Teistler
Lehrende:	Prof. Dr. M. Teistler, S. Reinhold, M.A., V. Schomakers, B.A.
Turnus:	im Wintersemester
Sprache:	deutsch
Schwerpunkt:	Human-Computer Interaction
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 120 h, Gesamtaufwand: 180 h
Leistungspunkte:	6 cp
Lernziele / Kompetenzen:	Sie kennen gängige Testmethoden, Testwerkzeuge und Testverfahren im Bereich Usability. Sie haben die Verfahren praktisch angewendet und können für verschiedene Anwendungen Usability-Tests konzipieren, planen und durchführen.
Inhalt:	<p>Vorlesung</p> <ul style="list-style-type: none"> • Theorie und Grundbegriffe (ISO-Norm, Einflussfaktoren und Ziele) • Usability im Software-Entwicklungsprozess (Zielgruppenanalyse, Personas, Prototyping, Informationsarchitektur, Szenarios) • Usability Testing (Thinking Aloud, Kontextanalyse, Eyetracking, Panelbefragung, Crowd-Testing, A/B Tests) • Usability-Metriken und -Richtlinien • Mobile Usability <p>Labor</p> <p>In den begleitenden Laboren konzipieren und planen Sie Usability-Tests für bestehende Software, führen die Tests im Usability-Labor durch und werten sie anschließend aus.</p>
Prüfungsform:	HA, Arb, Votr, K(2)
Literatur:	J. NIELSEN: Usability Engineering. Academic Press (1993) F. SARODNICK, H. BRAU: Methoden der Usability Evaluation. 2. Auflage, Huber (2011)