

Universität Paderborn



Informatik
**Veranstaltungs-
Kommentar**

Für

Informatik ▷ Bachelor
▷ Master

Computer Engineering

Ingenieurinformatik

Lehramt Informatik GyGe

Für das SoSe 18

Von der Fachschaft
Mathematik/Informatik



Inhaltsverzeichnis

1	Wichtige Informationen	3
1.1	Benutzerhinweise	3
1.2	Literaturangaben	3
1.3	Sprechstunden	3
1.4	Vollständigkeit	3
1.5	Internet	3
2	Mitarbeitende der Informatik	4
3	Prüfungssekretariate	8
4	Weitere wichtige Adressen	9
5	Veranstaltungen	10
5.1	Übersicht	10
5.2	Informatik	15
6	Raum für Notizen	44

Impressum

Herausgeber: Fachschaft Mathematik/Informatik
Universität Paderborn, Raum E1.311
Warburger Straße 100
33098 Paderborn
E-Mail: fsmi@uni-paderborn.de
Telefon: 05251 60-3260

V.i.S.d.P.: Jan Lippert

ISSN: 1868-0690

Redaktion: Alex Wiens, Jan Lippert, Oliver Rabe, Philipp Breuch

Mitarbeitende: die Fachschaft (Korrekturlesen),
die Dozentinnen und Dozenten der Mathematik und der Informatik (Kommentare)

Auflage: 2⁴ Exemplare

1 Wichtige Informationen

1.1 Benutzerhinweise

zum Kopf:

Name der Veranstaltung

Dozent: Name des Dozenten

Büro: Raum

Sprechstunde: Zeit

1.2 Literaturangaben

Die Bücher in diesem Abschnitt sind Empfehlungen der Dozenten. Viele dieser Bücher sind in der Bibliothek zu finden, sodass ihr euch die Bücher dort erst ansehen und ausleihen könnt, bevor ihr viel Geld dafür ausgeben müsst. Auf Ebene 3 der Bibliothek befindet sich übrigens der Seminarapparat unserer Fachschaft. In diesem haben wir etwas Grundlagenliteratur zur Informatik und Mathematik gesammelt, welche wir für lesenswert halten.

1.3 Sprechstunden

Ein Großteil der Dozentinnen und Dozenten gibt keine feste Sprechstunde mehr an, sondern ist nach Vereinbarung zu sprechen, sowie vor und nach den Veranstaltungen. Daher findet Ihr nicht überall die Angabe einer Sprechstunde.

1.4 Vollständigkeit

Da nicht alle Lehrenden einen Veranstaltungskommentar abgegeben haben, ist das Verzeichnis der Veranstaltungen leider nicht vollständig!

1.5 Internet

Elektronische Informationen zum Vorlesungsangebot gibt es unter folgenden Adressen:

- <https://cs.uni-paderborn.de/studium/studienangebot/informatik/> – offizielle Webseite zum Studienangebot der Informatik
- <https://math.uni-paderborn.de/studium/studiengaenge/mathematik/> – offizielle Webseite zum Studium der Mathematik
- <http://webptool.cs.upb.de/> – aktuellster Stand der Vorlesungsplanung
- <https://paul.upb.de/> – offizielles Vorlesungsverzeichnis der Uni

Die Seiten der Fachschaft findet Ihr hier: <http://die-fachschaft.de/>

2 Mitarbeitende der Informatik

Name	E-Mail	Telefon	Raum
Abshoff, Sebastian	abshoff@upb.de	6433	F1.125
Adelt, Peer	peer.adelt@c-lab.de	6051	FU.234
Affi, Haitham	haitham.affi@upb.de	5373	O3.167
Ahlers, Ulrich	uli@hni.upb.de	6700	F2.320
Allwright, Michael	michael.allwright@uni-paderborn.de	5465/202	ZM1.02-10
Amedick, Justin	justin.amedick@uni-paderborn.de	1727	O2.152
Anis, Anas	anas.anis@upb.de	3309	ZM1.02-11
Anjorin, Anthony	anthony.anjorin@upb.de	5465-224	ZM1.03-06
Antoniou, Paraskewi	antoniou@mail.upb.de	5394	O3.107
Anwer, Jahanzeb	jahanzeb.anwer@upb.de	4342	O3.128
Auroux, Sebastian	sebastian.auroux@upb.de	5385	O3.167
Bauer, Bernard	bb@upb.de	1737	O2.173
Benter, Markus	benter@upb.de	6454	F1.122
Benz, Manuel	manuel.benz@upb.de	3615	ZM1.02-07
Blobel, Johannes	blobel@ccs-labs.org	6494	F U.407
Blömer, Johannes	bloemer@upb.de	6651	F2.101
Blößl, Bastian	bloessl@ccs-labs.org	6494	F U.407
Blume, Bodo	bodo.blume@uni-paderborn.de	1750	O3.173
Bobolz, Jan	jan.bobolz@upb.de	6654	F2.111
Bodden, Eric	eric.bodden@upb.de	3313	ZM1.02-09
Borkowski, Philipp / Azubi	philipp.borkowski@upb.de	1727	O2.152
Boschmann, Alexander	alexander.boschmann@upb.de	5397	O3.119
Böttcher, Stefan	stb@upb.de	6662	F2.217
Brandt, Sascha	myeti@mail.upb.de	6451	F1.203
Brauer, Sascha	sbrauer@upb.de	6653	F2.106
Brunthaler, Stefan	s.brunthaler@upb.de	5380	O3.149
Budde, Lea	lea.budde@upb.de	6613	F2.201
Bunja, Kathrin	kabu@mail.upb.de	6627	F2.106
Bürger, Tanja	tanja.buerger@upb.de	5376	O3.155
Busa-Fekete, Robert,	busarobi@mail.upb.de	3348	O4.170
Buschmeyer, Carmen	carmen@upb.de	6412	F1.426
Buse, Dominik	buse@ccs-labs.org	6560	F1.416
Chvojka, Peter	chvojkap@mail.upb.de		F2.317
Dann, Andreas	andreas.dann@upb.de	5719	ZM1.02-07
Debessay, Seghid	seghid.debessay@c-lab.de	6166	FU.231
Divband Soorati, Mohammad	divband@mail.upb.de	6468	F1.213
Domik, Gitta	domik@upb.de	6610	F2.204
Drächsler, Sevil	sevil.mehraghdam@upb.de	1755	O3.161
Drees, Maximilan	maxdrees@mail.upb.de	6433	F1.125
Dressler, Falko	dressler@ccs-labs.org	6510	F1.401
Eidens, Fabian	feidens@msopb.de	6650	F2.108
Eikel, Benjamin	eikel@upb.de	6452	F1.203
Engels, Gregor	engels@upb.de	3337	ZM1.03.45
Feldkord, Björn	bjoernf@mail.uni-paderborn.de	6434	F1.125

Name	E-Mail	Telefon	Raum
Feldmann, Michael	michael.feldmann@upb.de	6722	F2.403
Feldmann, Rainer	obelix@upb.de	6720	F2.401
Feldotto, Matthias	feldi@mail.upb.de	6452	F1.203
Fischer, Holger	hfischer@s-lab.upb.de	5465/209	ZM1.03-37
Fischer, Matthias	mafi@upb.de	6466	F1.223
Geismann, Johannes	johannes.geismann@upb.de	3309	ZM1.02-11
Gerking, Christopher	christopher.gerking@upb.de	3307	ZM1.02-14
Gerstenberger, Dietrich	dietrich.gerstenberger@upb.de	6611	F2.201
Ghasemzadeh Mohammadi, Hassan	hgm@upb.de	4344	O3.134
Gmyr, Robert	gmyr@mail.upb.de	6704	F2.323
Graf, Tobias	tobias@mail.upb.de	1742	O3.128
Grieger, Marvin	grieger@mail.upb.de	5465/226	ZM1.03-12
Gundelach, Sigrid	sigu@upb.de	6655	F2.207
Hagenauer, Florian	hagenauer@ccs-labs.org	6492	F1.319
Hamann, Heiko	Heiko.Hamann@upb.de	6465	F1.221
Hartel, Rita	rst@upb.de	6612	F2.209
Hasnain, Asif	asif.hasnain@upb.de	1757	O3.164
Hauenschild, Wilfried	wilf@upb.de	5393	O2.146
Haupt, Jutta	jutta@upb.de	3312	ZM1.02.08
Heindorf, Stefan	heindorf@upb.de	5465-207	ZM1.03-07
Heinemann, Birte	birte.heinemann@upb.de	6615	F2.201
Henzgen, Sascha	shenzgen@mail.upb.de	3345	O4.164
Heppner, Sabrina	sheppner@mail.upb.de	6323	F2.209
Heyi, Binyam Shiferaw	binyam@mail.upb.de	1753	O3.161
Hinnenthal, Kristian	krijan@mail.upb.de	6732	F2.406
Ho, Nam	namh@mail.upb.de	5396	O3.122
Hottung, Andre	ahottenburg@mail.upb.de	26	ZM1.03-10
Hucke, Marion	marion.hucke@upb.de	6695	F2.411
Hüllermeier, Eyke	eyke@upb.de	1771	O4.258
Hülsmann, Adrian	klecks@upb.de	6623	F2.124
Jager, Tibor	tibor.jager@upb.de	6686	F2.308
Jahn, Claudia	claudia.jahn@uni-paderborn.de	6622	F2.104
Jakobs, Marie-Christine	mcjakobs@mai.upb.de	1767	O4.128
Jatzkowski, Jan	jan.jatzkowski@c-lab.de	6115	FU.313
Jazayeri, Bahareh	bahar.jazayeri@upb.de	5465-213	ZM1.03-11
Jovanovikj, Ivan	ijovanovikj@s-lab.upb.de	5465-216	ZM1.03-10
Juhnke, Jakob	jakob.juhnke@upb.de	6328	F2.108
Jung, Daniel	jungd@hni.uni-paderborn.de	6433	F1.125
Karl, Holger	hkarl@mail.uni-paderborn.de	5375	O3.158
Kastens, Uwe	uwe@upb.de	6686	F2.308
Kaufmann, Paul	paul.kaufmann@upb.de	1718	O3.116
Keil, Reinhard	Reinhard.Keil@uni-paderborn.de	6411	F1.428
Keller, Axel	kel@upb.de	1723	O2.149
Kemper, Michaela	mkemper@upb.de	1735	O2.164
Kenter, Tobias	kenter@upb.de	4340	O2.161
Kleine Büning, Hans	kbcs@upb.de	3360	O4.255
Kleinjohann, Bernd	bernd.kleinjohann@c-lab.de	6101	FU.214

Name	E-Mail	Telefon	Raum
Kleinjohann, Lisa	lisa.kleinjohann@c-lab.de	6102	FU.214
Klingler, Florian	klingler@ccs-labs.org	6493	F1.319
Kolb, Christina	ckolb@mail.upb.de	6725	F2.406
König, Jürgen	jkoenig@mail.upb.de	1715	O4.134
Kouchaksaraei, Hadi Razzaghi	hadi.razzaghi.kouchaksaraei@uni-paderborn.de	1758	O3.170
Krämer, Julia-Desirée	juliadk@mail.upb.de	5388	O4.131
Kraus, Hans-Joachim	hajo.kraus@upb.de	6538	F1.410
Krawinkel, Andreas	Krawi@upb.de	1725	O2.155
Kreiss, Lydia	Lydia.Kreiss@uni-paderborn.de	6501	F1.404
Krohn, Jörg-Peter	krohn@upb.de	3325	E1.103
Krüger, Jessica	jessica.krueger@upb.de	6614	F2.114
Krüger, Stefan	stefan.krueger@upb.de	3309	ZM1.02 -12
Kurek, Rafael	rafaeil.kurek@upb.de		
Lange, Christina	chlange@mail.upb.de	1749	O4.213
Lass, Michael	Michael.Lass@upb.de	1722	O2.149
Lengeling, Elisabeth	elisabeth.lengeling@upb.de	3361	O4.155
Lettmann, Theodor	lettman@upb.de	3350	O4.173
Li, Shouwei	sli@mail.uni-paderborn.de	6468	F1.216
Liske, Gennadij	utyf@mail.upb.de	6626	F2.111
Löken, Nils	nilo@upb.de	6632	F2.111
Lösch, Achim	achim.loesch@upb.de	5395	O3.122
Lücker, Steven Christopher	sluecker@mail.upb.de	6522	F1.107
Mäcker, Alexander	amaecker@mail.uni-paderborn.de	6427	F1.119
Magenheim, Johann S.	jsm@uni-paderborn.de	6341	F2.116
Malatyali, Manuel	malatya@upb.de	6462	F1.122
Maniera, Jürgen	sammy@upb.de	3326	ZM1.02.04
Margaritis, Melanie	melanie.margaritis@upb.de	6603	F2.201
Markarian, Christine	chrisss@mail.upb.de	6460	F1.225
Mehic, Ahmet	amehic@upb.de	3303	ZM1.02.12
Meisner, Sebastian	sebastian.meisner@upb.de	4347	O3.128
Melnikov, Vitaly	melnikov@mail.upb.de	3349	O4.158
Memedi, Agon	memedi@ccss.labs.org	6513	F1.416
Mergard, Claudia	claudia.mergard@upb.de	1727	O2.152
Meyer auf der Heide, F.	fmadh@upb.de	6480	F1.301
Mohr, Felix	fmohr@mail.upb.de	3352	O4.149
Mohsen, Ahmadi Fahander	ahamadim@mail.upb.de	3353	O4.167
	ahamadim@mail.upb.de	6513	F1.416
Monien, Burkhard	bm@upb.de	6707	F2.413
Mory, Martin	martin.mory@upb.de	5013	ZM1.02-13
Müller, Kathrin geb. Bröker	kathyb@uni-paderborn.de	6600	F2.416
Nabeel, Muhammad	nabeel@ccs-labs.org	6515	F 1.412
Neugebauer, Jonas	jonas.neugebauer@uni-paderborn.de	6611	F2.201
Ngonga Ngomo, Acel-Cyrille	axel.ngonga@upb.de	3342	O4.213
Niewöhner, Nadine	nadine.niewoehner@iem.fraunhofer.de	5465-403	ZM1.
Nitsche, Holger	hn@upb.de	1726	O2.155

Name	E-Mail	Telefon	Raum
Nouri, Zahra	znouri@mail.upb.de	5465-207	ZM1.03.07
Oberhoff, Andreas	oberhoff@upb.de	6514	F1.419
Oberhür, Simon	oberthür@upb.de	3844	ZM1.03-08
Ohrndorf, Laura	laura.orndorf@mail.upb.de	6613	F2.201
Parruca, Donald	donald.parruca@upb.de	1754	O3.164
Paul, Adil	adilp@mail.upb.de	3346	O4.161
Peuster, Manuel	manuel.peuster@upb.de	4341	O3.149
Pfahler, Peter	peter@upb.de	6688	F2.311
Pfannschmidt, Karlson	kiudee@mail.upb.de	3346	O4.161
Pierenkemper, Christoph	cpierenk@hni.upb.de		ZM1.
Platenius, Marie Christin	m.platenius@uni-paderborn.de	5465/176	ZM1.02-13
Platzner, Marco	platzner@upb.de	5250	O3.207
Plessl, Christian	christian.plessl@upb.de	5399	O3.110
	christain.plessl@upb.de	5399	O2.167
Podlipyan, Pavel	papo@mail.uni-paderborn.de	6460	F1.216
Rammig, Franz J.	franz@upb.de	6500	F1.414
Riebler, Heinrich	heirich.riebler@upb.de	5382	O2.158
Riechers, Sören	soerenri@mail.upb.de	6427	F1.119
Rittmeier, Florian	florian.rittmeier@s-lab.upb.de	3355	ZM1.03-37
Roger, Irene	irene.roger@upb.de	6620	F2.122
Rose, Sascha			ZM1
Saage, Sonja	ssaage@s-lab.upb.de	5582	ZM1.03-13
Sauer, Stefan	sauer@s-lab@upb.de	5390	ZM1.03-44
Schäfer, Johannes	jschaefer@mail.upb.de	6420	F1.119
Schäfer, Wilhelm	wilhelm@upb.de	2559	B2.326
Schäfermeyer, Petra	petral@upb.de	6481	F1.304
Scharfenbaum, Joachim	joscha@upb.de	3327	E1.101
Scheideler, Christian	scheideler@upb.de	6728	F2.326
Schlatt, Elisabeth	schlatt@mail.upb.de	3764	O4.125
Schneider, Stefan	stefan.schneider@upb.de	1753	O3.170
Schomaker, Gunnar	schomaker@sicp.de	5465-228	ZM1.03-14
Schönberg, Sven	sven.schoenberg@c-lab.de	6121	FU.234
Schroeder, Ulf-Peter	ups@upb.de	6726	F2.409
Schubert, Philipp	phillip.schubert@upb.de	5584	ZM1.02-11
Schulte, Carsten	carsten.schulte@upb.de	6343	F2.116
Schultz-Friese, Tobias	tsf@upb.de	6664	F2.224
Schwabe, Arne	arne.schwabe@upb.de	1756	O3.146
Schwichtenberg, Simon	schwicht@mail.upb.de	5290	ZM1.03-09
Selke, Harald	hase@upb.de	6413	F1.101
Senft, Björn	bjoern.senft@upb.de	5465-211	ZM1.03-08
Setzer, Alexander	asetzer@mail.upb.de	6721	F2.403
Shaker , Ammar	ammar.shaker@upb.de	3345	O4.164
Simon, Jens	simon@upb.de	1731	O2.164
Skopalik, Alexander	skopalik@mail.upb.de	6457	F1.209
Sommer, Christoph	sommer@ccs-labs.org	6495	F1.322
Stahl, Katharina	kasia@upb.de	5465-255	ZM1.03-37
Stoll, Christa	stoll@upb.de	3339	O4.213

Name	E-Mail	Telefon	Raum
Strothmann, Thim	thim@mail.upb.de	6730	F2.323
Szwillus, Gerd	szwillus@upb.de	6624	F2.119
Teetz, Alexander	alexandwer.teetz@upb.de	5465-215	ZM1.03-10
Thissen, Thomas	tici@upb.de	6701	F2.320
Töws, Manuel	mtoews@upb.de	4272	O4.122
Travkin, Oleg	oleg82@upb.de	1709	O4.131
Utermöhle, Michael	mike@upb.de	6666	F2.224
Vaz, Gavin	gavin.vaz@upb.de	1717	O2.158
von Rooijen, Lorijn	lorijn.van.rooijen@uni-paderborn.de	5713	ZM1.03-09
Wahby, Mostafa	mostafa.wahby@uni-paderborn.de	6468	F1.213
Wassing, Heinz-Georg,	koala@upb.de	6430	F1.122
Wawilow, Anastasia	apetkau@upb.de	6625	F2.124
Wegener, Friedhelm	fw@upb.de	3354	ZM1.03-04
Wehrheim, Heike	wehrheim@upb.de	4331	O4.225
Weskamp, Christoph	weskamp@dsor.de	5240	O2.466
Wiechers, Beatrix	wiechers@upb.de	3336	ZM1.03.46
Wiederhold, Cornelia	connyw@upb.de	6523	F1.101
Wiersema, Tobias	tobias.wiersema@upb.de	4343	O3.125
Wilke, Adrian	adrian.wilke@upb.de	6733	F2.416
Winkelkemper, Felix	winfel@hni.upb.de	6410	F1.104
Winnwa, Nils / Azubi	nils.winnwa@uni-paderborn.de	1727	O2.152
Wolters, geb. Bokermann, Dennis	dennis.bokermann@upb.de	5465-221	ZM1.03-07
Ygitbas, Enes	enes@upb.de	3356	ZM1.03-06
Zahn, Gero	gero.zahn@upb.de	6519	F1.107

3 Prüfungssekretariate

Name	E-Mail	Telefon	Raum
Manuel Leßmann Teamleitung MB & EIM Informatik/Computer Science Master Informatik Bachelor (Studierende, die den Bachelor im WiSe 2017/2018 abschließen werden) Ingenieurinformatik Schwerpunkt Elektrotechnik Bachelor Mathematik/Technomathematik Bachelor und Master	zps.eim1@zv.upb.de	5207	C2.332
Franziska Bartsch Computer Engineering Bachelor und Master (Nachnamen beginnend mit L-Z) Elektrotechnik Bachelor und Master (Nachnamen beginnend mit L-Z) Informatik Bachelor (Nachnamen beginnend mit L-Z) Wirtschaftsingenieurwesen Schwerpunkt Elektrotechnik Bachelor und Master	zps.eim2@zv.upb.de	4017	C2.216
Ramazan Uysal Computer Engineering Bachelor und Master (Nachnamen beginnend mit A-K) Elektrotechnik Bachelor und Master (Nachnamen beginnend mit A-K) Informatik Bachelor (Nachnamen beginnend mit A-K) Electrical Systems Engineering Master	zps.eim3@zv.upb.de	4230	C2.326

4 Weitere wichtige Adressen

Name	E-Mail	Telefon	Raum
Fachschaft Mathematik/Informatik Lernzentrum	fsmi@upb.de	3260	E1.311
Mathematik/Technomathematik		60-1856	J2.324
Mathe-Treff		3775	D3.331
Mathe-Lernzentrum		1856	J2.324
Rechnerbetreuung Didaktik	intermax@upb.de	3758	D2.339
Rechnerbetrieb Mathematik	pem@math.upb.de	3494	D2.301
Rechnerbetreuung Informatik	IRB-Support@upb.de	3318	E1.303

5 Veranstaltungen

5.1 Übersicht

Vorlesungen, für die uns bis Redaktionsschluss keine Kommentare erreicht haben, sind in der folgenden Übersicht mit -- gekennzeichnet.

Veranstaltungen, die bei Redaktionsschluss nicht mehr angeboten werden, sind im Inhaltsverzeichnis durchgestrichen.

Vorlesungen der Informatik

Grundstudium Bachelorstudiengang

Platzner	Digitaltechnik	(1.3)	--
Meyer auf der Heide	Datenstrukturen und Algorithmen	(1.2)	15
Sauer	Software Engineering	(1.1)	16
Keil	Grundlagen Mensch-Maschine-Wechselwirkung	(1.4)	18
Karl	Konzepte und Methoden der Systemsoftware	(1.3)	19
Bodden	Softwaretechnikpraktikum	(1.1)	20
Hartel	Grundlagen von Datenbanken	(1.1)	--

Hauptstudium Bachelorstudiengang

Algorithmen und Komplexität

Scheideler	Verteilte Algorithmen und Datenstrukturen	(2.1)	22
Blömer, Jager	Einführung in Kryptographie	(2.1)	23

Computersysteme

Platzner	Eingebettete Systeme	(3.1)	--
----------	----------------------	-------	----

Daten und Wissen

Ngonga Ngomo	Foundations of the Semantic Web	(4.1)	--
Hüllermeier	Grundlagen Wissensbasierter Systeme	(4.1)	24

Softwaretechnik

Wehrheim	Softwaremodellierung mit formalen Methoden	(1.1)	26
Pfahler	Programming Languages and Compilers	(1.1)	--
Anjorin	Modellbasierte Softwareentwicklung	(1.1)	27

Proseminare

Ngonga Ngomo	Building Knowledge Graphs	--	
Sommer	Kooperative Autonome Fahrzeuge	--	

Fischer	Algorithmische Geometrie		28
Wehrheim	Android App Analyse		30
Strothmann	SoftwareMakersLab: Design und Implementierung moderner HCI-Techniken		--
Jager	Sicherheit in Automatisierungstechnik und IoT		--

Masterstudiengang

Algorithm Design

Scheideler	Advanced Algorithms	(2.1,2.2,2.3)	31
Blömer	Foundations of Cryptography	(2.2,2.3)	32
Jager	Public-Key Cryptography	(2.3)	--
Fischer	Computational Geometry	(2.1,2.2)	33
Gharibian	Introduction to Quantum Computation	()	--

Computer Systems

Kleinjohann	Intelligence in Embedded Systems	(3.4,3.6)	--
Mohammadi	Algorithms for Synthesis and Optimization of Integrated Circuits	(3.4,3.5,3.6)	--
Simon	Architektur paralleler Rechnersysteme	()	34
Sommer	Cooperative Mobile Systems	(3.1,3.3,3.6)	--

Intelligence and Data

Kleinjohann	Intelligence in Embedded Systems	(3.4,3.6)	--
Ngonga Ngomo	Information Retrieval	(1.3)	--
Hüllermeier	Machine Learning II	(1.4)	35

Networks and Communication

Dressler	Network Simulation	(3.1,3.3,3.6)	--
Dressler	Vehicular Networking	(3.1,3.3,3.6)	--

Software Engineering

Bodden	Build It, Break It, Fix It	(1.5)	36
Bodden	Designing Code Analyses for Large-Scale Software	(1.1,1.5)	37

Seminare

Scheideler	PRISMA	(2.1,2.2)	--
Dressler	Selected Topics in Wireless Networking	(3.1,3.3,3.6)	--
Skopalik	Algorithmic Game Theory	(2.1,2.2)	--

Tauber	Assistive Technologies: Multi-modal Human-Computer Interaction	(4.1-4.3,4.5,4.6)	--
Domik	Current Topics in Visualization	(4.1-4.3,4.5,4.6)	--
Jager	Current Topics in IT Security	(2.3,3.1-3.3)	--

Projektgruppen

Projektgruppen die bereits im letzten Semester gestartet sind, werden nicht mit aufgeführt. Ob die Projektgruppe tatsächlich stattfindet, hängt von dem Verteilungsprozess ab.

Ngonga Ngomo	Data Extraction, Search, Analysis and Benchmarking II		--
Karl	Resource Management in Multi-Application Data Centres		--
Oberthür	Exploring Paderborn 7		--
Wever	On-The-Fly Machine Learning		--
Jager	Design, implementation and security analysis of a 0-RTT key exchange protocol		--
Nguyen	Automated Benchmark Management (ABM)		--
Mohammadi	EML: Embedded Machine Learning		--

Oberseminare

Hüllermeier	Maschinelles Lernen	(SWT&IS)	--
Wehrheim	Spezifikation und Modellierung	(SWT&IS)	--
Bodden	Softwaretechnik	(SWT&IS)	--
Engels	Informationssysteme	(SWT&IS)	--
Ngonga	Data and Web Science	(SWT&IS)	--
Scheideler	Theoretische Informatik	(MuA)	39
Meyer auf der Heide	Algorithmen und Komplexität	(MuA)	--
Blömer	Codes & Kryptographie	(MuA)	--
Meyer auf der Heide	SFB 901 On-The-Fly Computing	(MuA)	--
Domik	Interactive 3D Graphics and Games	(MuA/MMWW)	--
Karl	Rechnernetze	(ESS)	--
Platzner	Technische Informatik	(ESS)	--
Dressler	Distributed Embedded Systems	(ESS)	--
Platzner	Approximate Computing	(ESS)	--
Plessl	Hochleistungs-IT-Systeme	(ESS)	--
Sommer	Kooperative Fahrzeugsysteme	(ESS)	--
Jager	IT-Sicherheit	(ESS)	--

Schulte	Informatik und Bildung	(MMWW)	--
Keil	Informatik und Gesellschaft	(MMWW)	--
Domik	Computergrafik, Visualisierung und Bildverarbeitung	(MMWW)	--

Didaktik der Informatik

Schulte	Fachdidaktische Grundlagen		--
Schulte	Softwarepraktikum		--
Schulte	Fachdidaktische Konzepte		--
Schulte	Stufenbezogene Unterrichtsmodelle		--

Lehrveranstaltungen für andere Studiengänge

Selke	Einführung in die Informatik für Geisteswissenschaftler		40
Lettmann	Webbasierte Informationssysteme		--
Selke	Gesellschaft und Informationstechnik		41

Allgemeine Veranstaltungen

Gerstenberger, Schulte	Agile Organisation von Lerngruppen		42
Gerstenberger, Schulte	Evaluation des Workloads von Lerngruppen		43
Plesl	PC ² Kolloquium: Kolloquium der Parallelverarbeitung		--

5.2 Informatik

Datenstrukturen und Algorithmen

Dozent: Meyer auf der Heide

Büro: F1.301

Sprechstunde: Mi, 13-14 Uhr

Inhaltsangabe

Algorithmen bilden die Grundlage jeder Hardware und Software: ein Schaltkreis setzt einen Algorithmus in Hardware um, ein Programm macht einen Algorithmus für den Rechner verständlich. Algorithmen spielen daher eine zentrale Rolle in der Informatik. Wesentliches Ziel des Algorithmenentwurfs ist die (Ressourcen-)Effizienz, d.h. die Entwicklung von Algorithmen, die ein gegebenes Problem möglichst schnell und mit möglichst geringem Speicherplatz lösen.

Untrennbar verbunden mit effizienten Algorithmen sind effiziente Datenstrukturen, also Methoden, große Datenmengen im Rechner so zu organisieren, dass Anfragen wie Suchen, Einfügen und Löschen, aber auch komplexere Anfragen effizient beantwortet werden können.

Die in dieser Veranstaltung vorgeschlagenen Entwurfs- und Analysemethoden für effiziente Algorithmen und Datenstrukturen sowie die grundlegenden Beispiele wie Sortierverfahren, dynamische Datenstrukturen und Graphenalgorithmien gehören zu den Grundlagen für die Algorithmenentwicklung und Programmierung in weiten Bereichen der Informatik.

Ab Sommersemester 2019 gibt es in dieser Veranstaltung erstmalig ein Praktikum. Der genaue Ablauf und die Organisation werden auf der Webseite der Veranstaltung sowie in der ersten Vorlesung bekannt gegeben.

Literaturangaben

- **Cormen, Leiserson, Rivest, Stein:** *Introduction to Algorithms*, MIT Press / McGraw-Hill, 3rd ed., ISBN: 0-262-53305-8
- **Cormen, Leiserson, Rivest:** *Algorithmen - Eine Einführung*, Oldenburg, ISBN: 3-486-27515-1
- **Ottmann, Widmeyer:** *Algorithmen und Datenstrukturen*, Spektrum Akademischer Verlag, ISBN: 3-321-29535-8
- **Kleinberg, Tardos:** *Algorithm Design*, Addison-Wesley, ISBN: 0-312-29535-8
- **Sedgewick:** *Algorithms in Java (parts 1-4)*, Addison-Wesley, ISBN: 0-201-36120-5

Verschiedenes

Hörerkreis:

i-b2, i-l2, ie2, im2, winf2

vorausgesetzte Kenntnisse:

keine

nächster Wiederholungstermin:

SS 2019

Modulzugehörigkeit:

Info 1. Studienabschnitt, MUA

weiterführende Veranstaltungen:

Grundlegende Algorithmen, Methoden des Algorithmenentwurfs, Verteilte Algorithmen und Datenstrukturen

Homepage:

<http://www.hni.upb.de/alg/lehre>

Software Engineering

Dozent: Sauer

Büro: ZM1.03-44 / FU.237

Sprechstunde: nach Vereinbarung

Inhaltsangabe

In der Vorlesung werden die Grundlagen der systematischen und ingenieurmäßigen Softwareentwicklung vermittelt. Im Fokus steht dabei die modellbasierte Softwareentwicklung.

Die Vorlesung führt in wesentliche Vorgehensmodelle für die Softwareentwicklung ein, sowohl klassische als auch agile. Es werden Methoden für die Softwareentwicklung und -qualitätssicherung vermittelt, die innerhalb der Vorgehensmodelle zum Einsatz kommen.

Außerdem werden Modellierungssprachen und Softwarewerkzeuge vorgestellt, mit denen die statischen und dynamischen Aspekte von Softwaresystemen beschrieben werden können. Insbesondere wird die objektorientierte Modellierungssprache UML (Unified Modeling Language) eingeführt, die unterschiedliche Diagrammsprachen wie Klassendiagramme, Komponentendiagramme, Use-Case-Diagramme, Aktivitätendiagramme, Sequenzdiagramme und Zustandsdiagramme vereint. Modellierungswerkzeuge werden exemplarisch eingesetzt.

Die Vorlesung wird abgerundet durch eine durchgängige Entwicklungsmethode von der Anforderungsspezifikation über den Architektur- und Softwareentwurf bis hin zur Implementierung und dem Testen der Software. Hierbei wird vor allem auf die Aspekte der systematischen Ableitung und Verfeinerung von Modellen, der Transformation von Modellen in Programmcode (Codegenerierung) sowie des modellbasierten Testens eingegangen. Es werden methodische Hinweise zur Erstellung der Ergebnisartefakte (u.a. Richtlinien, Architekturstile und Entwurfsmuster) und zur Prüfung ihrer Qualität sowie zum Einsatz der Modellierungssprachen im Softwareentwicklungsprozess gegeben.

Darüber hinaus werden Techniken zur Definition und domänenspezifischen Anpassung von Modellierungssprachen (Metamodellierung, UML-Profile sowie Beispiele konkreter domänenspezifischer Sprachen (DSLs) wie SysML oder BPMN) betrachtet.

Die Vorlesung wird durch Übungen begleitet, in denen die Vorlesungsinhalte aufgegriffen, vertieft und an beispielhaften Entwicklungsaufgaben selbst angewendet werden.

Praktikum: Begleitend zur Vorlesung Software Engineering werden an einem durchgängigen Beispiel ausgehend von der Spezifikation von Anforderungen sowohl die Modellierung von Softwaresystemen als auch der Übergang von den Modellen zur Implementierung sowie zum modellbasierten Testen der Softwaresysteme bearbeitet.

Verschiedenes

Hörerkreis:

Informatik Bachelor, Computer Engineering
Bachelor, Wirtschaftsinformatik Bachelor,
Ingenieurinformatik, Informatik Lehramt

Modulzugehörigkeit:

Software Engineering

Prüfungsform:

Klausur

Leistungspunkte:

5 ECTS

weiterführende Veranstaltungen:

Softwaretechnikpraktikum

nächster Wiederholungstermin:

jeweils im Sommersemester

Homepage:

[https://cs.uni-paderborn.de/dbis/
lehre/veranstaltungen/ss-2018/
software-engineering/aktuelles/](https://cs.uni-paderborn.de/dbis/lehre/veranstaltungen/ss-2018/software-engineering/aktuelles/)

Grundlagen Mensch-Maschine Wechselwirkung

Dozent: Keil

Büro: F1.428

Sprechstunde: n. V.

Inhaltsangabe

Die Gebrauchstauglichkeit von Software ist heute eine Kernanforderung an die Produktgestaltung in der Informatik. Dabei geht es zum einen darum mit Hilfe geeigneter Gestaltungsmaßnahmen Verständnisprozesse bei Nutzern zu fördern und unnötige Belastungen bei der Arbeit mit Softwaresystemen zu vermeiden. Zum anderen ist ein methodisches Repertoire erforderlich, um schon während des Entwurfs die Gebrauchstauglichkeit sichern zu können. Dazu sind eine Reihe von fachlichen Grundlagen erforderlich, die von rechtlichen Anforderungen über physiologische und psychologische Grundlagen bis hin zu Methoden und Techniken der Systemgestaltung reichen.

Die Studierenden sollen in die Lage versetzt werden, grundlegende Problembereiche der Mensch-Rechner-Interaktion zu erkennen und sie konstruktiv gestaltend umzusetzen. Damit sollen sie zugleich anschlussfähiges Wissen erwerben, das vor allem für die Zusammenarbeit mit Designern und Ergonomen erforderlich ist, aber auch für den Diskurs mit Medienwissenschaftlern und Pädagogen hilfreich ist. Die erworbenen Kenntnisse und Fertigkeiten bilden zugleich die Grundlage für vertiefende Veranstaltungen im Bereich der Mensch-Maschine-Wechselwirkung wie z.B. Usability Engineering, Webdesign oder Assistierende Technologien, Barrierefreiheit.

Literaturangaben

Wird in der Veranstaltung bekanntgegeben.

Verschiedenes

Modulzugehörigkeit:

I.4.1 Grundlagen Mensch Maschine-
Wechselwirkung

vorausgesetzte Kenntnisse:

Grundsätzliche Kenntnisse über die Entwicklung von Softwaresystemen.

nächster Wiederholungstermin:

SS 19

Homepage:

koala.uni-paderborn.de

Konzepte und Methoden der Systemsoftware

Dozent: Karl

Büro: O3.158

Sprechstunde: Montag, 13 - 14 Uhr

Inhaltsangabe

Die Veranstaltung KMS betrachtet grundlegende Funktionen systemnaher Software, insbesondere die wesentlichen Funktionen, die ein Betriebssystem erbringen muss. Eine besondere Herausforderung ist dabei die Behandlung von Nebenläufigkeit, die in vielfältiger Form in modernen IT-Systemen auftritt. Grundlegende Techniken wie Isolation und Virtualisierung werden wir auf unterschiedlichen Abstraktionsebenen vorfinden und wiederholt anwenden.

Ziel der Vorlesung ist die Vermittlung solch allgemeiner Prinzipien, Konzepte, Methoden und Techniken, wie sie in komplexen Hardware/Software-Systemen mit Nebenläufigkeit vorzufinden sind. Die Studenten sollen die Gemeinsamkeiten erkennen können und die Prinzipien als grundlegend für das Fach verstehen. Sie sollen insbesondere in Entwurfssituationen diese Methoden sinnvoll einsetzen können.

Literaturangaben

Es gibt kein Buch, das alle in der Vorlesung behandelten Themen sinnvoll abdeckt. Empfehlenswert ist jedoch die aktuelle Ausgabe von:

- **Stallings, W.:** *Betriebssysteme: Prinzipien und Umsetzung*, Pearson Studium

Verschiedenes

Hörerkreis:

Bachelor

nächster Wiederholungstermin:

SS 2019

Modulzugehörigkeit:

Modul I.3.2

Homepage:

<https://cs.uni-paderborn.de/cn/teaching/courses/konzepte-und-methoden-der-systemsoftware/>

Softwaretechnikpraktikum

Dozent: Bodden

Büro: F1.125

Sprechstunde: Nach Absprache

Inhaltsangabe

Das Softwaretechnikpraktikum ist eine praxisorientierte Lehrveranstaltung inklusive Vorlesungen zum Thema Projektmanagement. Eine komplexe Softwareentwicklungsaufgabe wird im Team von ca. zehn Studierenden unter Verwendung von UML und Java bearbeitet. Schwerpunkt des Praktikums ist die Vermittlung von Erfahrungen mit der gruppenorientierten Softwareentwicklung unter Benutzung marktüblicher Werkzeuge, Methoden und Prozesse. Die Erstellung von Meilensteinplänen, ein teilweise durch die Studierenden zu übernehmendes Projektmanagement, sowie die Anfertigung von Kostenschätzungen und die Protokollierung des Aufwandes durch Stundenzettel sind wesentliche Bestandteile, um die Praxisnähe des Projekts sicherzustellen.

Literaturangaben

- **Helmut Balzert:** *Lehrbuch der Software-Technik: Basiskonzepte und Requirements Engineering*, Heidelberg: Spektrum, 2009
- **Helmut Balzert:** *Lehrbuch der Software-Technik: Entwurf, Implementierung, Installation und Betrieb*, Heidelberg: Spektrum, 2011
- **Helmut Balzert:** *Lehrbuch der Software-Technik: Softwaremanagement*, Heidelberg: Spektrum, 2008
- **Daniel Galin:** *Software Quality Assurance: From theory to implementation*, Harlow: Pearson
- **Watts S. Humphrey:** *Three Dimensions of Process Improvement Part I: Process Maturity*, CrossTalk: The Journal of Defense Software Engineering, Februar 1998, <http://static1.1.sqspcdn.com/static/f/702523/9167925/1288288753533/199802-Humphrey.pdf>
- **Watts S. Humphrey:** *Three Dimensions of Process Improvement Part II: The Personal Process*, CrossTalk: The Journal of Defense Software Engineering, März 1998, <http://static1.1.sqspcdn.com/static/f/702523/9167977/1288288919673/199803-Humphrey.pdf>
- **Watts S. Humphrey:** *Three Dimensions of Process Improvement Part III: The Team Process*, CrossTalk: The Journal of Defense Software Engineering, April 1998, <http://static1.1.sqspcdn.com/static/f/702523/9168059/1288289124207/199804-Humphrey.pdf>
- **Ian Sommerville:** *Software Engineering*, Harlow: Pearson

Verschiedenes

Hörerkreis:

Informatik Bachelor, Ingenieurinformatik
Bachelor mit Schwerpunkt Informatik

Prüfungsform:

Projektarbeit, Klausur

vorausgesetzte Kenntnisse:

Teilnahmevoraussetzung für das Softwaretechnikpraktikum ist ein erfolgreicher Prüfungsabschluss in den Veranstaltungen Grundlagen der Programmierung 1 u. 2 sowie Softwareentwurf.

nächster Wiederholungstermin:

Wintersemester 2018

Homepage:

<https://www.hni.uni-paderborn.de/swt/lehre/softwaretechnik-softwarepraktikum-ss-2016>

Modulzugehörigkeit:

I.1.2 Softwaretechnik

Leistungspunkte:

8 ECTS

weiterführende Veranstaltungen:

Modellbasierte Softwareentwicklung

Vorbesprechung:

erster Vorlesungstermin

Verteilte Algorithmen und Datenstrukturen

Dozent: Scheideler

Büro: F2.326

Sprechstunde: Do, 16-17 Uhr

Inhaltsangabe

Die Vorlesung wird eine Einführung in die Grundlagen der verteilten Algorithmen und Datenstrukturen geben. Folgende Themen werden dabei behandelt:

1. Netzwerktheorie
2. Designprinzipien für verteilte Algorithmen und Datenstrukturen
3. Einführung in die verteilte Programmierung
4. Prozessorientierte Datenstrukturen ((zyklische) Listen, Clique und hypercubische Graphen)
5. Informationsorientierte Datenstrukturen (verteilt Hashing, verteilte Stacks und Queues und verteilte Heaps)

Dabei wird ein besonderer Wert auf Datenstrukturen und Algorithmen gelegt, die selbst für sehr große verteilte Systeme noch effizient arbeiten und die selbststabilisierend sind.

Literaturangaben

Für die Vorlesung werden wir auf neueste Forschungsergebnisse zurückgreifen. Dadurch gibt es keine Bücher, die begleitend zur Vorlesung gelesen werden können, aber zur Vorlesung wird es einen ausführlichen Foliensatz geben.

Verschiedenes

Hörerkreis:

i-b6, ie, im, winf

Prüfungsform:

Mündliche Prüfung und Softwareprojekt

vorausgesetzte Kenntnisse:

Datenstrukturen und Algorithmen

weiterführende Veranstaltungen:

Fortgeschrittene verteilte Algorithmen und Datenstrukturen

Vorbesprechung:

-

Modulzugehörigkeit:

Info 2. Studienabschnitt, MuA

Leistungspunkte:

6 ECTS

nützliche Parallelveranstaltungen:

-

nächster Wiederholungstermin:

SS 2019

Homepage:

<http://cs.uni-paderborn.de/ti/lehre/veranstaltungen/ss-2018/vads/>

Einführung in Kryptographie

Dozent: Blömer

Büro: F2.101

Sprechstunde: Mi, 13 - 14 Uhr

Inhaltsangabe

Die Kryptographie beschäftigt sich mit Methoden zur sicheren Datenübertragung. Moderne Kryptographie ist eine Schlüsseltechnologie mit vielen Anwendungen, von der EC-Karte, Mobiltelefon, TV-Decodern und elektronischem Geld bis zur fälschungssicheren elektronischen Unterschrift auf Bestellungen und Verträgen im Internet.

In dieser Vorlesung werden einige der grundlegenden modernen Kryptosysteme wie AES und RSA vorgestellt. Weiter werden die wichtigsten Sicherheitskonzepte der modernen Kryptographie diskutiert.

Literaturangaben

- **Buchmann:** *Einführung in die Kryptographie*, 5. Auflage, Springer, 2010. ISBN: 978-3-642-11185-3
- **Katz, Lindell:** *Introduction to Modern Cryptography*, Chapman & Hall / CRC Press, 2007. ISBN: 1-5848-8551-3
- **Stinson:** *Cryptography: Theory and Practice*, 2nd edition, Chapman & Hall / CRC Press, 2001. ISBN: 1-5848-8206-9

Verschiedenes

Hörerkreis:

Bachelor Informatik

Modulzugehörigkeit:

Informatik, Gebiet Algorithmen und Komplexität

Prüfungsform:

je nach Teilnehmerzahl Klausur oder mündliche Prüfung

Leistungspunkte:

6 ECTS

vorausgesetzte Kenntnisse:

Lineare Algebra, nützlich sind auch Kenntnisse in Algebra und Zahlentheorie

weiterführende Veranstaltungen:

Foundations of Cryptography

nächster Wiederholungstermin:

voraussichtlich in einem Jahr

Grundlagen Wissensbasierter Systeme

Dozent: Hüllermeier

Büro: O4.258

Sprechstunde: nach Vereinbarung

Inhaltsangabe

Intelligente Systeme sind Computersysteme, deren Verhalten durch Methoden und Algorithmen der Künstlichen Intelligenz (KI) gesteuert wird. Solche Systeme gewinnen kontinuierlich an Bedeutung, nicht nur auf wissenschaftlichen Ebene sondern auch im sozialen und gesellschaftlichen Kontext: Autonome oder teilautonome Systeme wie Serviceroboter, selbstfahrende PKWs oder medizinische Diagnosesysteme werden unser privates und berufliches Leben in absehbarer Zukunft tiefgreifend verändern. Diese Vorlesung gibt eine Einführung in Methoden und Konzepte der Künstlichen Intelligenz. Der inhaltliche Schwerpunkt liegt dabei auf Wissensbasierten Systemen im Sinne von Systemen, die mithilfe adäquater Ansätze zur Repräsentation, Akquisition und Verarbeitung von Wissen die Problemlösungskompetenz eines Fachexperten in einer bestimmten Anwendungsdomäne approximieren. Neben Methoden der Wissensrepräsentation und -verarbeitung gibt die Vorlesung auch einen ersten Einblick in den automatisierten Erwerb von Wissen mithilfe maschineller Lernverfahren.

Struktur der Vorlesung:

- Komponenten wissensbasierter Systeme
- Logische Grundlagen und Wissensrepräsentation
- Regelbasierte Inferenz
- Modellierung von Unsicherheit und Vagheit
- Graphische Modelle und probabilistische Inferenz
- Einführung in das Maschinelle Lernen

Literaturangaben

- **S. Russell and P. Norvig.:** *Artificial Intelligence: A Modern Approach*, 3. Auflage, Pearson, 2010
- **C. Beierle und G. Kern-Isberner:** *Methoden wissensbasierter Systeme*, 4. Auflage, Vieweg+Teubner, 2008
- **P. Flach:** *Machine Learning*, Cambridge Univ. Press, 2012
- **M.M. Richter and R.O. Weber:** *Case-Based Reasoning*, Springer, 2013
- **R. Kruse, C. Borgelt, F. Klawonn, C. Moewes, G. Ruß, M. Steinbrecher:** *Computational Intelligence*, Vieweg+Teubner, 2011

Verschiedenes

Hörerkreis:

Informatik, Wirtschaftsinformatik, Mathematik

Modulzugehörigkeit:

Daten und Wissen

Prüfungsform:

Klausur

Leistungspunkte:

6 ECTS

nächster Wiederholungstermin:

Sommersemester 2018/19

Homepage:

<https://cs.uni-paderborn.de/is/>

Softwaremodellierung mit formalen Methoden

Dozent: Wehrheim

Büro: O4.225

Sprechstunde: nach Vereinbarung

Inhaltsangabe

Formale Methoden sind Sprachen zur Modellierung/Spezifikation von Systemen. Ein Modell eines (Soft- oder Hardware) Systems beschreibt auf einer gewissen Abstraktionsebene die Funktionalität des Systems. Im Gegensatz zu (den meisten) Programmiersprachen besitzen formale Methoden eine genau festgelegte Semantik, d.h. eine mathematische Beschreibung der Bedeutung einer Spezifikation. Diese Festlegung der Semantik erlaubt es, das Systemmodell bereits vor der eigentlichen Implementierung formal zu analysieren und mögliche Fehler frühzeitig zu finden.

In der Vorlesung sollen verschiedene formale Methoden eingeführt werden, die für unterschiedliche Systemarten geeignet sind. Für jede dieser formalen Methoden werden Semantik und Analysetechniken vorgestellt und Modellierungsbeispiele zur Illustration des Einsatzbereiches besprochen.

Am Anfang der Vorlesung wird es vorrangig um die Modellierung von Parallelität und Kommunikation gehen. Hier werden Petrinetze und die Prozessalgebra CCS vorgestellt. Danach werden wir uns mit Sprachen zur Beschreibung von zeitlichen Aspekten (Timed Automata) und mit zustandsbasierten Formalismen zur Spezifikationen von Daten und Operationen (Z und Object-Z) beschäftigen.

Verschiedenes

Hörerkreis:

Bachelor Informatik, 2. Studienabschnitt

Modulzugehörigkeit:

SWT

Prüfungsform:

Klausur oder mündliche Prüfung

Leistungspunkte:

6 ECTS

vorausgesetzte Kenntnisse:

Prädikatenlogik, etwas Lineare Algebra, formale Beweise

nächster Wiederholungstermin:

SS 2020

Homepage:

<http://cs.uni-paderborn.de/sms/teaching/>

Modellbasierte Softwareentwicklung (MBSE)

Dozent: Anjorin

Büro: ZM1.03-06

Sprechstunde: Mo, 10-11 Uhr

Inhaltsangabe

Die Studierenden sollen grundlegende Verfahren zur Konstruktion großer Softwaresysteme kennen und ihre Anwendung beherrschen. Sie sollen die Vor- und Nachteile von Spezifikationstechniken erfahren, die Notwendigkeit von Design erkennen und Modelle zur Verbesserung der Softwarequalität einsetzen können. Unter anderem wird auf das Paradigma der „Model Driven Development“ eingegangen, das einen wesentlichen Produktivitäts- und Qualitätsgewinn bei der Softwareentwicklung verspricht.

Inhalt (unter anderem):

- DSL-Engineering
- Metamodellierung
- Code-Generierung
- Modelltransformationen
- Modellgetriebene Entwicklungsprozesse
- aktuelle Trends und Forschungsthemen

Verschiedenes

Hörerkreis:

Bachelor Informatik

Leistungspunkte:

6 ECTS

vorausgesetzte Kenntnisse:

Grundlagen der Programmierung (1+2),
Softwareentwurf

weiterführende Veranstaltungen:

Fundamentals of Model-Driven Engineering
(FMDE)

nächster Wiederholungstermin:

kontinuierlich

Proseminar: Algorithmische Geometrie

Dozent: Fischer

Büro: F1.223

Sprechstunde: nach Vereinbarung

Inhaltsangabe

Im Proseminar werden Vorträge zu Methoden aus der Algorithmischen Geometrie angeboten. Die Vorträge werden aus unterschiedlichen Büchern zusammengestellt:

Computational Geometry: Algorithms and Applications, Mark de Berg, Otfried Cheong, Marc van Kreveld, Mark Overmars, Springer-Verlag, 2008, <https://dx.doi.org/10.1007/978-3-540-77974-2>

Computational Geometry: An Introduction, Michael I. Shamos, Franco P. Preparata, Springer-Verlag, 1985, <https://dx.doi.org/10.1007/978-1-4612-1098-6>

Das de Berg Buch ist von Rechnern der Uni Paderborn aus zugreifbar.
Es werden unter anderem behandelt:

- Schnitt von Liniensegmenten
- Polygontriangulierung
- Lineares Programmieren
- Bereichsuche
- Voronoi Diagramme und Delaunay Triangulierungen
- Konvexe Hüllen
- BSP Bäume
- Sichtbarkeitsgraphen
- Quadtrees
- Bewegungsplanung

Dozenten:

Matthias Fischer, Ulf-Peter Schroeder

Literaturangaben

- **Mark de Berg, Otfried Cheong, Marc van Kreveld, Mark Overmars:** *Computational Geometry*, Springer-Verlag, 2008, <https://dx.doi.org/10.1007/978-3-540-77974-2>
- **Michael I. Shamos, Franco P. Preparata:** *Computational Geometry: An Introduction*, Springer-Verlag

Verschiedenes

Hörerkreis:

Bachelor Informatik

Prüfungsform:

Seminarvortrag und Ausarbeitung

Vorbesprechung:

am ersten Veranstaltungstag

Modulzugehörigkeit:

2. Studienabschnitt des BSc-Studiengangs

vorausgesetzte Kenntnisse:

Datenstrukturen und Algorithmen ist von Vorteil

Homepage:

<http://www.hni.uni-paderborn.de/alg/lehre>

Proseminar Android App Analysis

Dozent: Wehrheim

Büro: O4.225

Sprechstunde: nach Vereinbarung

Inhaltsangabe

Android stellt heute das am weitesten verbreitete Betriebssystem für mobile Endgeräte dar. Nicht zuletzt deswegen ist es wichtig sich mit den Sicherheitskonzepten dieses Betriebssystems auseinanderzusetzen, um z.B. zu erkennen, welche Anwendungen bzw. Apps als sicherheitskritisch einzustufen sind.

Im Verlauf des Seminars werden wir uns sowohl die hauseigenen Sicherheitskonzepte von Android, als auch wissenschaftliche Projekte wie beispielsweise Software Analyse Tools für Android genauer ansehen.

Verschiedenes

Hörerkreis:

inf-b5

Modulzugehörigkeit:

übergreifend

Prüfungsform:

Vortrag, aktive Beteiligung, schriftliche Ausarbeitung

vorausgesetzte Kenntnisse:

Programmierkenntnisse

Vorbesprechung:

erscheint auf Webseite

Homepage:

<http://cs.uni-paderborn.de/sms/teaching/>

Fortgeschrittene Algorithmen (Advanced Algorithms)

Dozent: Scheideler

Büro: F2.326

Sprechstunde: Do, 16-17 Uhr

Inhaltsangabe

Diese Vorlesung wird sich mit fortgeschrittenen Algorithmen vor allem im Bereich der randomisierten Algorithmen beschäftigen. Dabei werden folgende Themen behandelt:

1. Randomisierte Datenstrukturen
2. Randomisiertes Runden
3. Wahrscheinlichkeitsamplifikation
4. Randomisierte Metrikreduktion
5. Niedrigdimensionale Optimierungsprobleme
6. Sublineare Algorithmen

Literaturangaben

- **R. Motwani and P. Raghavan:** *Randomized Algorithms*, Cambridge University Press, 1995
- **N. Alon and J. Spencer:** *The Probabilistic Method*, John Wiley, 1991

Verschiedenes

Hörerkreis:

i-m

Modulzugehörigkeit:

Info 3. Studienabschnitt, MUA

Prüfungsform:

Mündliche Prüfung

Leistungspunkte:

6 ECTS

vorausgesetzte Kenntnisse:

Datenstrukturen und Algorithmen, Grundlagen in Wahrscheinlichkeitstheorie

nächster Wiederholungstermin:

unklar

Homepage:

<http://cs.uni-paderborn.de/ti/lehre/veranstaltungen/ss-2018/aa>

Foundations of Cryptography

Dozent: Blömer

Büro: F2.101

Sprechstunde: Mi, 13 - 14 Uhr

Inhaltsangabe

We discuss several advanced security concepts for encryption schemes, digital signatures, and identification protocols. These concepts include semantic security, indistinguishable encryptions, existential unforgeability, and zero-knowledges. We also show how these properties of cryptographic primitives can be achieved from general assumptions like the existence of one-way functions or trapdoor permutations. Finally, we compare the security of cryptographic primitives used in practice (like AES and RSA) with advanced security concepts.

Literaturangaben

- **O. Goldreich:** *Foundations of Cryptography I+II*, Cambridge University Press
- **Katz, Lindell:** *Introduction to Modern Cryptography*, Chapman & Hall/CRC

Verschiedenes

Hörerkreis:

Computer Science Master Course

Modulzugehörigkeit:

Focus area: Algorithm Design, Networks and Communication

Prüfungsform:

oral exam

Leistungspunkte:

6 ECTS

nächster Wiederholungstermin:

SS2019

Computational Geometry

Dozent: Fischer

Büro: F1.223

Sprechstunde: nach Vereinbarung

Inhaltsangabe

The lecture provides topics of Computational Geometry. Topics include:

Voronoi diagrams, Epsilon networks and VC-dimension, algorithmic motion planning for robots, visibility in polygons, convex hull, lower contour of line segments and functions, Sweep method and applications, geometric data structures: dynamization, kd-trees, range trees, priority search tree.

De Berg's, Matousek, and Klein's book can be accessed free of charge from the university's computers.

Literaturangaben

- **Mark de Berg, Otfried Cheong, Marc van Kreveld, Mark Overmars:** *Computational Geometry*, Springer-Verlag, 2008, <https://dx.doi.org/10.1007/978-3-540-77974-2>
- **Jiri Matousek:** *Lectures on Discrete Geomtetry*, Springer-Verlag, 2002, <https://dx.doi.org/10.1007/978-1-4613-0039-7>
- **Michael I. Shamos, Franco P. Preparata:** *Computational Geometry: An Introduction*, Springer-Verlag, 1985, <https://dx.doi.org/10.1007/978-1-4612-1098-6>
- **Rolf Klein:** *Algorithmische Geometrie*, Springer-Verlag, 2005, <https://dx.doi.org/10.1007/3-540-27619-X>

Verschiedenes

Hörerkreis:

Informatik Master

Prüfungsform:

oral examination

Homepage:

<https://www.hni.upb.de/alg/teaching>

Modulzugehörigkeit:

III.2.1, III.2.2, Focus Area: Algorithm Design

vorausgesetzte Kenntnisse:

Advanced understanding of data structures and algorithms. For example the lecture Data Structures and Algorithms of Bachelor's studies.

Architektur paralleler Rechnersysteme

Dozent: Simon

Büro: O2.161

Inhaltsangabe

1. Programmierparadigmen
2. Shared-Memory Architekturen
3. Distributed-Memory Architekturen
4. High-Performance-Computer (HPC) Architekturen

Verschiedenes

Hörerkreis:

Informatik Master, Serviceveranstaltung für
Computer Engineering

Modulzugehörigkeit:

MSc CS + CE

Prüfungsform:

mündliche Prüfung

Leistungspunkte:

6 ECTS

vorausgesetzte Kenntnisse:

Grundlagen Rechnerarchitektur. Grundlagen
Programmiersprachen.

nützliche Parallelveranstaltungen:

High-Performance Computing (Prof. Plesl)

Homepage:

<https://pc2.uni-paderborn.de/teaching/lectures/architektur-paralleler-rechnersysteme-hpc-architectures-sole-2018/>

nächster Wiederholungstermin:

SoSe 2019

Machine Learning II (in English)

Dozent: Hüllermeier

Büro: O4.258

Sprechstunde: nach Vereinbarung

Inhaltsangabe

Machine learning is “the art and science of algorithms that make sense of data” (Flach, 2012). Due to the ever increasing amount of data that is routinely produced in our information society, the topic of machine learning has become increasingly important in the recent years, not only as a scientific discipline but also as a key technology of modern software and information systems.

This lecture will cover selected topics in contemporary machine learning research, such as reinforcement learning, online learning and bandit algorithms, multi-task learning, multi-target and structured output prediction, preference learning, learning from weak supervision, and uncertainty in machine learning. The focus of the lecture will be on methods and algorithms, though theoretical issues and applications will be addressed, too.

Literaturangaben

- **Y.S. Abu-Mostafa, M. Magdon-Ismail, H.T. Lin:** *Learning from Data*, AMLBook, 2012
- **P. Flach:** *Machine Learning*, Cambridge Univ. Press, 2012
- **E. Alpaydin:** *Machine Learning*, Oldenbourg, 2008
- **C.M. Bishop:** *Pattern Recognition and Machine Learning*, Springer, 2006

Verschiedenes

Hörerkreis:

(Ing.-)Informatik Master, Winfo Master

Leistungspunkte:

6 ECTS

nächster Wiederholungstermin:

Summer term 2017/2018

Modulzugehörigkeit:

Modul III.1.4 (SWT&IS Wissensbasierte Systeme, Kleine Büning)

vorausgesetzte Kenntnisse:

This lecture is a continuation of the "Machine Learning I" course. It assumes elementary knowledge of learning from data and model induction (theory of generalization), including methods for basic (supervised) learning tasks.

Homepage:

<https://cs.uni-paderborn.de/is/>

Build It, Break It, Fix It

Dozent: Bodden

Büro: F1.125

Sprechstunde: Nach Absprache

Inhaltsangabe

This course aims on learning basic principles of secure software development in a very practical fashion. Students will be ask to gather in teams and develop small software projects based on a formal specification, also including security requirements. In a second iteration, the developed software will be exchanged between development teams to break the implementation, i.e., find and exploit security vulnerabilities, of other teams. Afterwards, teams will get the chance to fix the found vulnerabilities and, hence, render their software product secure. The course will contain small theoretical parts in which basic strategies of secure software development and software exploitation are presented. Nevertheless, the course is of very practical nature and since securing a software product, as well as breaking it, demands a wide variety of skills and creativity, quite high amounts of motivation and self organization are required.

Literaturangaben

- Ruef, Andrew and Hicks, Michael and Parker, James and Levin, Dave and Mazurek, Michelle L. and Mardziel, Piotr: *Build It, Break It, Fix It: Contesting Secure Development*, CCS 16

Verschiedenes

Hörerkreis:

Informatik Master

Modulzugehörigkeit:

III.1.5 Analytische Methoden des Software Engineering

Prüfungsform:

Phasenbezogene Prüfungsleistung

Leistungspunkte:

4 ECTS

vorausgesetzte Kenntnisse:

A mature understanding of the Java programming language and software security requirements. Secure software development practices will be helpful.

nützliche Parallelveranstaltungen:

Secure Software Engineering

nächster Wiederholungstermin:

WiSe 2018

Designing code analyses for large-scale software

Dozent: Bodden

Büro: F1.125

Sprechstunde: Nach Absprache

Inhaltsangabe

Statische Codeanalysen dienen dazu, automatisiert Fehler und Schwachstellen im Programmcode aufzufinden. Zu diesem Zwecke suchen sie nach bekannten Fehlermustern. In dieser Vorlesung wird erklärt, wie man solche Codeanalysen entwirft, die inter-prozedural sind, also das komplette Programm betrachten, über die Grenzen einzelner Prozeduren hinweg. Der Entwurf solcher Analysen gestaltet sich deshalb sehr schwierig, weil die Analysen oft Millionen von Programmstatements gleichermaßen präzise aber auch effizient verarbeiten müssen. Es werden außerdem Beispielsanalysen aus dem Bereich der IT-Sicherheit besprochen.

Behandelte Themen:

1. Intra-prozedurale Datenflussanalyse
2. Algorithmen zur Konstruktion von Call-Graphen
3. Kontext-insensitive inter-prozedurale Datenflussanalyse
4. Kontext-Sensitivität mittels des Call-Strings-Ansatzes
5. Value-basierte Kontexte
6. Kontext-Sensitivität mittels des Funktionalen Ansatzes
7. Distributive Probleme effizient mit den IFDS- und IDE-Frameworks lösen
8. Aktuelle Herausforderungen in der inter-prozeduralen Datenflussanalyse

Während der gesamten Veranstaltung werden Anwendungsbeispiele aus dem Gebiet der Softwaresicherheit diskutiert.

Literaturangaben

- **Thomas Reps, Susan Horwitz, and Mooly Sagiv.:** *dataflow analysis via graph reachability. POPL 95*, 1995. Precise interprocedural
- **Shmuel Sagiv, Thomas W. Reps, and Susan Horwitz.:** *Dataflow Analysis with Applications to Constant Propagation. TAPSOFT 95*, 1995. Precise Interprocedural
- **Akash Lal, Thomas Reps, and Gogul Balakrishnan.:** *2005. Extended weighted pushdown systems.*, CAV 2005
- **Nomair A. Naeem, Ondrej Lhoták, and Jonathan Rodriguez.:** *2010. Practical extensions to the IFDS algorithm.*, CC 2010
- **Yannis Smaragdakis, Martin Bravenboer, and Ondrej Lhoták:** *2011. Pick your contexts well: understanding object-sensitivity.*, POPL 2011
- **Eric Bodden.:** *2012. Inter-procedural data-flow analysis with IFDS/IDE and Soot.*, SOAP 2012
- **Rohan Padhye, Uday P. Khedker.:** *Interprocedural Data Flow Analysis in Soot using Value Contexts.*, SOAP 2013

Verschiedenes

Hörerkreis:

Master Informatik

Modulzugehörigkeit:

III.1.1 Modellbasierte Softwareentwicklung,
III.1.5 Analytische Methoden des Software
Engineering

Prüfungsform:

mündliche Prüfung

Leistungspunkte:

6 ECTS

vorausgesetzte Kenntnisse:

Der vorherige Besuch der Veranstaltung Software Analysis wird empfohlen ist aber nicht zwingend erforderlich. Ein gutes Verständnis von Java und den Prinzipien objektorientierter Programmierung ist hilfreich.

nächster Wiederholungstermin:

Wintersemester 2018

Homepage:

[https://www.hni.uni-paderborn.de/
swt/lehre/deca](https://www.hni.uni-paderborn.de/swt/lehre/deca)

Oberseminare

Oberseminar: Theoretische Informatik

Dozent: Scheideler

Büro: F2.326

Sprechstunde: Do, 16-17 Uhr

Inhaltsangabe

Im Oberseminar werden neueste Forschungsergebnisse im Bereich der theoretischen Informatik vorgestellt. Sie richtet sich in erster Linie an interessierte Master Studenten und wissenschaftliche Mitarbeiter.

Verschiedenes

Hörerkreis:

i-m

nächster Wiederholungstermin:

WS 2018

vorausgesetzte Kenntnisse:

Fortgeschrittenes Wissen im Bereich der theoretischen Informatik.

Homepage:

<http://cs.uni-paderborn.de/ti/lehre/oberseminar/ss-2018>

Einführung in die Informatik für Geisteswissenschaftler

Dozent: Selke

Büro: F1.101

Sprechstunde: nach Vereinbarung

Inhaltsangabe

Die Studierenden erhalten einen Einblick in die Denk- und Arbeitsweisen der Informatik. Sie lernen Basiskonzepte und -begriffe der verschiedenen Teilbereiche der Informatik kennen und erwerben ein Basisverständnis bezüglich Formalisierung sowie die Fähigkeit, Software nicht mehr von der Nutzer- sondern von der Erstellerseite her zu denken. Die Studierenden gewinnen einen Einblick in die gesellschaftliche Bedeutung der Informatik und der digitalen Medien sowie in die Fachsprache der Informatik und damit die Fähigkeit, zwischen verschiedenen Fachsprachen und Fachkulturen zu moderieren.

Die Veranstaltung umfasst unter anderem folgende Inhalte: Von Schrift und Zahl zu Hypermedia, Grundlagen des Internet, Darstellung von Daten in Computern, Trennung von Form und Inhalt am Beispiel HTML, Formale Beschreibung von Regelsystemen, Algorithmen, Kernkonzepte höherer Programmiersprachen, Einführung in die Programmierung am Beispiel JavaScript, Abstrakte Datenstrukturen.

Verschiedenes

Hörerkreis:

Medienwissenschaften (BA), mewi, mi2, mm

Prüfungsform:

Klausur, Mini-Projekt, aktive Teilnahme an den Übungen

weiterführende Veranstaltungen:

Einführung in Web-bezogene Sprachen

Vorbesprechung:

Vorlesungstermin in der ersten Semesterwoche

Modulzugehörigkeit:

Modul 9: Einführung in die Informatik für Geisteswissenschaftler

Leistungspunkte:

8 ECTS

nächster Wiederholungstermin:

Sommer 2019

Homepage:

koala.uni-paderborn.de

Gesellschaft und Informationstechnik

Dozent: Selke

Büro: F1.101

Sprechstunde: nach Vereinbarung

Inhaltsangabe

Die Veranstaltung gibt einen Überblick über die gesellschaftlichen Aspekte der Informationstechnik und versetzt die Studierenden in die Lage, die gesellschaftliche und ethische Bedeutung des Fachs zu beurteilen und Konsequenzen für verantwortungsbewusstes Handeln zu ziehen. Die Studierenden verstehen die Zusammenhänge zwischen Informationstechnik und Gesellschaft, sind in der Lage, die Auswirkungen informationstechnischer Produkte und Dienstleistungen zu analysieren und potentielle Konfliktfelder zu identifizieren, und kennen ethische Verhaltensregeln, wie sie zum Beispiel von den Berufsorganisationen IEEE und ACM herausgegeben werden.

Die Veranstaltung umfasst unter anderem folgende Inhalte: Geschichte der Informatik, Einführung in Ethik, Kommunikation in Netzwerken, Geistiges Eigentum, Datenschutz, Sicherheit von Computern und Netzwerken, Zuverlässigkeit von Computern, Professionelle Verantwortung, Wechselwirkungen zwischen Informatik und Gesellschaft.

Literaturangaben

- **Michael J. Quinn:** *Ethics for the Information Age*, 7th edition, Pearson Education, 2015

Verschiedenes

Hörerkreis:

Computer Engineering (BA)

Prüfungsform:

Klausur

nützliche Parallelveranstaltungen:

Rechtliche Grundlagen für IT-Berufe

Vorbesprechung:

Vorlesungstermin in der ersten Semesterwoche

Modulzugehörigkeit:

Modul Recht und Gesellschaft

Leistungspunkte:

3 ECTS

nächster Wiederholungstermin:

Winter 2018/2019

Homepage:

koala.uni-paderborn.de

Agile Organisation von Lerngruppen

Dozent: Gerstenberger

Büro: F2.201

Sprechstunde: nach Vereinbarung

Inhaltsangabe

Zusammen mit den Studierenden wird eine agile Planungsweise (angelehnt an Scrum) erarbeitet, mit der der Studienalltag organisiert werden soll. Die studentische Lerngruppe übernimmt dabei die Rolle des Developer Teams, das interdisziplinär und selbstorganisiert seine Arbeiten plant und umsetzt. Während der Umsetzung werden die Zeitbudgets und die Ergebnisse aus den Übungszetteln und Klausuren anonym erfasst und evaluiert.

Im Rahmen dieser Veranstaltung sollen der Arbeitseinsatz und Studienerfolg der Lerngruppen evaluiert werden. Voraussetzung dafür wäre die Teilnahme jeweils vollständiger Lerngruppen. Bei der Anmeldung zu dieser Veranstaltung wird versucht dies sicherzustellen, bzw. es könnte auch eine Option sein, im Rahmen der Anmeldung Lerngruppen zu bilden.

Verschiedenes

Hörerkreis:

B.Sc. Informatik, B: Sc. Computer Engineering, B.Ed. HRSGe, B.Ed. BK, B.Ed. Gy/Ge

Prüfungsform:

Studienleistung: Aktive Teilnahme und Gestaltung der Sprint Planning-, Daily Standup- und Sprint Review-Meetings. Teilnahme an der Gruppenarbeit und der semesterbegleitenden Evaluation.

Leistungspunkte:

4 ECTS im Studium Generale

Evaluation des Workloads von Lerngruppen

Dozent: Gerstenberger

Büro: F2.201

Sprechstunde: nach Vereinbarung

Inhaltsangabe

Die Veranstaltung dient der anonymen Erfassung von Zeitbudgets und Ergebnissen aus Übungszetteln und Klausuren von unbetreuten Lerngruppen als Vergleich zu den Daten aus der Veranstaltung „Agile Organisation (des Studienalltags) von Lerngruppen“.

Im Rahmen dieser Veranstaltung sollen der Arbeitseinsatz und Studienerfolg von Lerngruppen evaluiert werden. Voraussetzung dafür wäre die Teilnahme jeweils vollständiger Lerngruppen. Bei der Anmeldung zu dieser Veranstaltung wird versucht dies sicherzustellen, bzw. es könnte auch eine Option sein, im Rahmen der Anmeldung Lerngruppen zu bilden.

Verschiedenes

Hörerkreis:

B.Sc. Informatik, B: Sc. Computer Engineering, B.Ed. HRSGe, B.Ed. BK, B.Ed. Gy/Ge

Prüfungsform:

Studienleistung: Aktive Teilnahme an der semesterbegleitenden Evaluation.

Leistungspunkte:

1 ECTS im Studium Generale

vorausgesetzte Kenntnisse:

keine

6 Raum für Notizen

Stundenplan

Uhrzeit	Montag	Dienstag	Mittwoch	Donnerstag	Freitag
7 - 8					
8 - 9					
9 - 10					
10 - 11					
11 - 12					
12 - 13					
13 - 14					
14 - 15					
15 - 16					
16 - 17					
17 - 18					
18 - 19					
19 - 20					