

## Inhalt

|   |    |
|---|----|
| <i>Gebietszuordnung</i> .....   | 3  |
| <i>Vertiefung Bildverarbeitung</i> .....                                    | 8  |
| <i>Vertiefung Computergraphik und Visualisierung</i> .....                  | 8  |
| <i>Vertiefung Information Systems Engineering</i> .....                     | 9  |
| <i>Vertiefung Optimierung</i> .....   | 10 |
| <i>Vertiefung Theoretische Informatik</i> .....                             | 12 |
| <i>Vertiefung Wissenschaftliches Rechnen</i> .....                          | 12 |
| <i>Module aus der Informatik</i> .....                                      | 14 |
| Algorithmische Geometrie (IAGM) .....                                       | 14 |
| Computergraphik 1 (ICG1) .....  | 15 |
| Computergraphik 2 (ICG2) .....  | 16 |
| Compilerbau (ICOM) .....  | 17 |
| Computerspiele (ICS).....   | 18 |
| Foundations of Cryptography and Network Security (ICNS).....                | 19 |
| Data Mining - Algorithmen und parallele Verfahren (IDM).....                | 21 |
| Data Warehouses (IDW).....  | 22 |
| Effiziente Algorithmen 1 (IEA1).....  | 24 |
| Effiziente Algorithmen 2 (IEA2).....  | 25 |
| Software-Praktikum für Fortgeschrittene (IFM) .....                         | 26 |
| Formale Sprachen und Automatentheorie (IFSA) .....                          | 28 |
| Flow and Transport in Terrestrial Systems (IFTTS) .....                     | 29 |
| Grundlagen der wissenschaftlichen Visualisierung (IGWV) .....               | 30 |
| Inverse Problems (IIP) .....  | 32 |
| Knowledge Discovery in Databases (IKDD) .....                               | 33 |
| Künstliche Intelligenz (IKI) .....  | 34 |
| Konsistenzmodelle für parallele und verteilte Systeme (IKPV).....           | 35 |
| Logik in der Informatik (ILI).....  | 36 |
| Masterarbeit (IMa) .....  | 37 |
| Management und Analyse von Datenströmen (IMADS).....                        | 38 |
| Mixed-Integer Programming and Combinatorial Optimization (IMIP) .....       | 40 |
| Numerik von Transportprozessen in porösen Medien (INTPM) .....              | 41 |
| Objekterkennung und automatisches Bildverstehen (IOAB).....                 | 42 |
| Objektorientiertes Programmieren im Wissenschaftlichen Rechnen (IOPWR)..... | 43 |
| Paralleles Höchstleistungsrechnen (IPHR).....                               | 44 |
| Parallele Lösung großer Gleichungssysteme (IPLGG).....                      | 46 |
| IT-Projektmanagement (IPM).....   | 47 |
| Räumliche Datenbanken (IRDB) .....  | 48 |
| Seminar (IS).....   | 50 |
| Simulationswerkzeuge (ISIMW).....   | 51 |
| Software Evolution (ISWEvol).....   | 52 |
| Qualitätsmanagement (ISWQM) .....   | 54 |
| Requirements Engineering (ISWRE) .....                                      | 55 |
| Visualisierung im Bereich Cultural Heritage (IVCH) .....                    | 56 |
| Verteilte Datenbanken und Informationssysteme (IVDB) .....                  | 57 |
| Verteilte Systeme I (IVS1) .....  | 58 |
| Visualisierung in Natur- und Technikwissenschaften (IVNT) .....             | 60 |
| Volumenvisualisierung (IVV) .....   | 61 |

|  |    |
|--|----|
| Wissenschaftliches Arbeiten (IWA) .....  | 63 |
| <i>Module, die auch im Bachelor/Master Mathematik angeboten werden:</i> .....  | 64 |
| Mathematische Logik (MB9) .....  | 64 |
| Wahrscheinlichkeitstheorie (MC4).....  | 65 |
| Numerik (MD1) .....  | 66 |
| Statistik (MD2).....   | 67 |
| Lineare Optimierung (MD3).....   | 68 |
| Nichtlineare Optimierung (MD4) .....   | 69 |
| Wissenschaftliches Rechnen (MD5).....  | 69 |
| Computeralgebra I (MG19) .....   | 70 |
| Computeralgebra II (MG20) .....  | 72 |
| Numerische Lineare Algebra (MH5) .....   | 72 |
| Numerik gewöhnlicher Differentialgleichungen (MH6) .....   | 73 |
| Numerik partieller Differentialgleichungen (MH7) .....   | 74 |
| Numerische Optimierung bei Differentialgleichungen (MH8) .....   | 75 |
| Statistik II (MH12) .....  | 76 |
| Wahrscheinlichkeitstheorie II (MH13) .....   | 77 |
| Berechenbarkeit und Komplexität I (MH14) .....   | 78 |
| Berechenbarkeit und Komplexität II (MH15).....   | 79 |
| Algorithmische Optimierung I (MH16).....   | 80 |
| Algorithmische Optimierung II (MH17).....  | 81 |
| Mustererkennung (MH18).....  | 81 |
| Bildverarbeitung (MH19) .....  | 82 |
| <i>Module, die im Master Technische Informatik angeboten werden:</i> .....   | 83 |
| <i>Module, die auch im Bachelor/ Master Physik angeboten werden</i> .....  | 83 |
| Physics of Imaging (MWInf5) .....  | 84 |
| Image Processing (MWInf6).....   | 84 |
| Pattern Recognition (MWInf7).....  | 86 |
| Introduction to Image Processing on the GPU (IGPU).....  | 87 |
| Fast Parallel Implementations of Image Labeling Problems (IMLP) .....  | 88 |
| <i>Module, die auch im Bachelor/ Master Biologie angeboten werden:</i> .....   | 90 |
| Model-based segmentation of biomedical images (Modellbasierte Segmentierung<br>Biomedizinischer Bilder, IMSBI) ..... | 90 |
| <i>Module im Anwendungsgebiet:</i> .....   | 91 |
| Anwendungsgebiet (IAG).....  | 91 |

Im Folgenden sind die Module des Masterstudiengangs Angewandte Informatik beschrieben. Es können (aber müssen nicht) Vertiefungen gewählt werden, dabei sind die nachfolgenden Anmerkungen zu beachten.

Wie in der Prüfungsordnung beschrieben, sind bei der Wahl der Module Gebiet abzudecken. Die Zuordnung der Module zu den Gebieten ist nachfolgend beschrieben..

### **Gebietszuordnung**

Entsprechend der in der Prüfungsordnung genannten Vorgaben ist eine Auswahl aus den folgenden Gebieten abzudecken.

Bildverarbeitung (BV)

Computergraphik und Visualisierung (CGV)

Datenbanksysteme (DB)

Diskrete und Kombinatorische Optimierung (Opt)

Parallele und Verteilte Systeme (PVS)

Software Engineering (SWE)

Technische Informatik (Tecl)

Theoretische Informatik (Theol)

Wissenschaftliches Rechnen (WR)

Die nachfolgende Tabelle definiert die Gebietszuordnung der im Modulhandbuch beschriebenen Module. Alle nicht genannten Module sind keinem speziellen Gebiet zugeordnet.

| Modul  | B<br>V | C<br>G<br>V | D<br>B | O<br>p<br>t | P<br>V<br>S | S<br>W<br>E | T<br>e<br>c<br>l | Th<br>e<br>o<br>l | W<br>R |
|--|--------|-------------|--------|-------------|-------------|-------------|------------------|-------------------|--------|
| Algorithmische Geometrie (IAGM)                  |        | x           |        |             |             |             |                  |                   |        |
| Computergraphik 1 (ICG1)                         |        | x           |        |             |             |             |                  |                   |        |
| Computergraphik 2 (ICG2)                         |        | x           |        |             |             |             |                  |                   |        |
| Computerspiele (ICS)                             |        | x           |        |             |             |             |                  |                   |        |
| Foundations of Cryptography and Network Security |        |             |        |             | x           |             |                  |                   |        |
| Datenbanken 2 (IDB2)                             |        |             | x      |             |             |             |                  |                   |        |
| Data Mining (IDM)                                |        |             |        |             | x           |             |                  |                   |        |
| Data Warehouses (IDW)                            |        |             | x      |             |             |             |                  |                   |        |
| Effiziente Algorithmen 1 (IEA1)                  |        |             |        | x           |             |             |                  |                   |        |

|   |  |   |   |   |   |  |  |   |   |
|---|--|---|---|---|---|--|--|---|---|
| Effiziente Algorithmen 2 (IEA2)                                 |  |   |   | x |   |  |  |   |   |
| Formale Sprachen und Automatentheorie (IFSA)                    |  |   |   |   |   |  |  | x |   |
| Flow and Transport in Terrestrial Systems (IFTTS)               |  |   |   |   |   |  |  |   | x |
| Grundlagen der wissenschaftlichen Visualisierung (IGWV)         |  | x |   |   |   |  |  |   |   |
| Inverse Problems (IIP)  |  | x |   |   |   |  |  |   |   |
| Knowledge Discovery in Databases (IKDD)                         |  |   | x |   |   |  |  |   |   |
| Künstliche Intelligenz (IKI)                                    |  | x |   |   |   |  |  |   |   |
| Konsistenzmodelle für parallel und verteilte Systeme (IKPV)     |  |   |   |   | x |  |  |   |   |
| Logik in der Informatik (ILI)                                   |  |   |   |   |   |  |  | x |   |
| Management und Analyse von Datenströmen (IMADS)                 |  |   | x |   |   |  |  |   |   |
| Mixed-Integer Programming and Combinatorial Optimization (IMIP) |  |   |   | x |   |  |  |   |   |
| Netzwerk-analyse  |  |   |   |   |   |  |  |   | x |
| Numerik von Transportprozessen in porösen Medien (INTPM)        |  |   |   |   |   |  |  |   | x |
| Objektorientiertes Programmieren im Wissenschaftliche           |  |   |   |   |   |  |  |   | x |

|   |  |   |   |  |   |   |  |   |   |
|---|--|---|---|--|---|---|--|---|---|
| n Rechnen (IOPWR)   |  |   |   |  |   |   |  |   |   |
| Paralleles Höchstleistungsrechnen (IPHR)  |  |   |   |  |   |   |  |   | x |
| Parallele Lösung großer Gleichungssysteme (IPLGG)                                   |  |   |   |  |   |   |  |   | x |
| IT-Projektmanagement (IPM)  |  |   |   |  |   | x |  |   |   |
| Räumliche Datenbanken (IRDB)  |  |   | x |  |   |   |  |   |   |
| Simulationswerkzeuge (ISIMW)  |  |   |   |  |   |   |  |   | x |
| Sicherheit in Rechnersystemen (ISIR)  |  |   |   |  | x |   |  |   |   |
| Moderne Architekturen: Komponentenbasierte und serviceorientierte Systeme (ISWArch) |  |   |   |  |   | x |  |   |   |
| Qualitätsmanagement (ISWQM)   |  |   |   |  |   | x |  |   |   |
| Requirements Engineering (ISWRE)  |  |   |   |  |   | x |  |   |   |
| Visualisierung im Bereich Cultural Heritage (IVCH)                                  |  | x |   |  |   |   |  |   |   |
| Verteilte Datenbanken und Informationssysteme (IVDB)                                |  |   | x |  |   |   |  |   |   |
| Visualisierung in Natur- und Technikwissenschaften (IVNT)                           |  | x |   |  |   |   |  |   |   |
| Verteilte Systeme I (IVS1)  |  |   |   |  | x |   |  |   |   |
| Volume Visualization (IVV)  |  | x |   |  |   |   |  |   |   |
| Mathematische   |  |   |   |  |   |   |  | x |   |

|  |   |  |  |   |  |  |  |   |   |
|--|---|--|--|---|--|--|--|---|---|
| Logik (MB9)  |   |  |  |   |  |  |  |   |   |
| Numerik (MD1)  |   |  |  |   |  |  |  |   | x |
| Lineare Optimierung (MD3)                                |   |  |  | x |  |  |  |   |   |
| Nichtlineare Optimierung (MD4)                           |   |  |  | x |  |  |  |   |   |
| Wissenschaftliches Rechnen (MD5)                         |   |  |  |   |  |  |  |   | x |
| Numerische Lineare Algebra (MH5)                         |   |  |  |   |  |  |  |   | x |
| Numerik Gewöhnlicher Differentialgleichungen (MH6)       |   |  |  |   |  |  |  |   | x |
| Numerik Partieller Differentialgleichungen (MH7)         |   |  |  |   |  |  |  |   | x |
| Numerische Optimierung bei Differentialgleichungen (MH8) |   |  |  | x |  |  |  |   |   |
| Numerische Methoden der Kontinuumsmechanik (MH9)         |   |  |  | x |  |  |  |   |   |
| Spezielle Themen der Numerik (MH10)                      |   |  |  | x |  |  |  |   |   |
| Spezielle Themen der Optimierung (MH11)                  |   |  |  | x |  |  |  |   |   |
| Berechenbarkeit und Komplexität I (MH14)                 |   |  |  |   |  |  |  | x |   |
| Berechenbarkeit und Komplexität II (MH15)                |   |  |  |   |  |  |  | x |   |
| Algorithmische Optimierung I (MH16)                      |   |  |  | x |  |  |  |   |   |
| Algorithmische Optimierung II (MH17)                     |   |  |  | x |  |  |  |   |   |
| Physics of Image Processing (MWInf5)                     | x |  |  |   |  |  |  |   |   |

|  |   |  |  |  |  |  |   |  |  |
|--|---|--|--|--|--|--|---|--|--|
| Bildverarbeitung<br>(MH19 oder<br>MWInf6)  | x |  |  |  |  |  |   |  |  |
| Statistische<br>Mustererkennung<br>(MH18 oder<br>MWInf7)                                     | x |  |  |  |  |  |   |  |  |
| Objekterkennung<br>und<br>automatisches<br>Bildverstehen<br>(IOAB)                           | x |  |  |  |  |  |   |  |  |
| Modellbasierte<br>Segmentierung<br>biomedizinischer<br>Bilder (IMSBI)                        | x |  |  |  |  |  |   |  |  |
| Introduction to<br>Image Processing<br>on the GPU<br>(IGPU)                                  | x |  |  |  |  |  |   |  |  |
| Fast Parallel<br>Implementations<br>of Image Labeling<br>Problems (IMLP)                     | x |  |  |  |  |  |   |  |  |
| Alle Grundlagen-<br>und Vertiefungs-<br>Module aus dem<br>Master<br>Technische<br>Informatik |   |  |  |  |  |  | x |  |  |

## **Vertiefung Bildverarbeitung**

Die Vertiefung wird gestaltet vom Heidelberger Zentrum für Bildverarbeitung (Heidelberg Collaboratory Image Processing). Sie kann im Master gewählt werden. Sie umfasst 2 Seminare, Wahlpflichtveranstaltungen und die Masterarbeit. Es sind 2 Seminare und vertiefende Vorlesungen aus dem Gebiet der Bildverarbeitung zu wählen und die Masterarbeit in diesem Gebiet anzufertigen. Weitere Module in der Informatik sowie das Anwendungsgebiet können gemäß der Prüfungsordnung frei gewählt werden.

Die Veranstaltungen aus dem Gebiet der Bildverarbeitung sind

Physics of Image Processing (MWInf5)  
Bildverarbeitung (MWInf6 oder MH19)  
Statistische Mustererkennung (MWInf7 oder MH18)  
Objekterkennung und automatisches Bildverstehen (IOAB)  
Multidimensionale Signalanalyse (TIMDS)

Dabei sind die Module MWInf6 und MWInf7 mehr auf die Physik ausgerichtet und die Module MH18 und MH19 mehr auf die Mathematik. Sie zählen jeweils 8 LP.

| Veranstaltung                                      | Semester | LP  |
|--|----------|-----|
| Wiss. Arbeiten                                     | 1        | 2   |
| Seminar – Bildverarbeitung                         | 1-3      | 4   |
| Seminar – Bildverarbeitung                         | 1-3      | 4   |
| Bildverarbeitung (MWInf6 oder MH19)                | 1-3      | 8   |
| Statistische Mustererkennung (MWInf7 oder MH18)    | 1-3      | 8   |
| Weitere Module aus dem Gebiet der Bildverarbeitung | 1-3      | 20  |
| Weitere Module aus mindestens 2 anderen Gebieten   | 1-3      | 26  |
| Masterarbeit Bildverarbeitung                      | 4        | 30  |
| Anwendungsgebiet                                   | 1-3      | 18  |
| LP Summe   |          | 120 |

## **Vertiefung Computergraphik und Visualisierung**

Diese Vertiefung befähigt zur Entwicklung von Algorithmen und Anwendungsprogrammen für die visuelle Datenverarbeitung und -analyse. Hierzu gehören Kenntnisse im Bereich Mensch-Maschine-Interaktion, Computergraphik, Datenanalyse und wissenschaftliche Visualisierung. Die Vertiefung kann im Bachelor, im Master und durchgängig vom Bachelor zum Master gewählt werden. Sie umfasst mindestens ein Seminar, ein Praktikum, die Masterarbeit und zusätzliche Wahlpflichtveranstaltungen. Die Module sind aus dem Lehrgebiet Computergraphik und Visualisierung (CGV) zu wählen. Weitere Veranstaltungen sollte aus zwei ergänzenden Lehrgebieten gewählt werden.

Vertiefende Module aus dem Gebiet CGV sind  
 IAGM (4 LP): Algorithmische Geometrie  
 ICG1 (6 LP): Computergraphik 1  
 ICG2 (6 LP): Computergraphik 2  
 ICS (8 LP): Computerspiele  
 IGWV (8 LP): Grundlagen der wissenschaftlichen Visualisierung  
 IKI (6 LP): Künstliche Intelligenz  
 IVV (8 LP): Volume Visualization  
 IVNT (4 LP): Visualisierung in Natur- und Technikwissenschaften  
 IVCH (2 LP): Visualisierung im Bereich Cultural Heritage

Für das ergänzende Lehrgebiet bieten folgende Bereiche eine gute Erweiterung.

- Bildverarbeitung
- Datenbanken
- Optimierung
- Software Engineering
- Wissenschaftliches Rechnen

Weitere Bereiche können auf Antrag genehmigt werden.

| Veranstaltung   | Semester | LP  |
|---|----------|-----|
| Wiss. Arbeiten  | 1        | 2   |
| Seminar – Lehrgebiet CGV                                  | 1-3      | 4   |
| Seminar – CGV oder ergänzendes Lehrgebiet                 | 1-3      | 4   |
| Fortgeschrittenen-Praktikum CGV                           | 1-3      | 8   |
| Vertiefende Module aus dem Bereich CGV                    | 1-3      | 20  |
| Weitere Module aus mindestens 2 weiteren Gebieten und CGV | 1-3      | 34  |
| Masterarbeit CGV  | 4        | 30  |
| Anwendungsgebiet  | 1-3      | 18  |
| LP Summe  |          | 120 |

Bei Belegung der Vertiefung im Bachelor und Master müssen die im Master belegten Module disjunkt zu denen im Bachelor sein.

### ***Vertiefung Information Systems Engineering***

Die Vertiefung kann im Bachelor, im Master und durchgängig vom Bachelor zum Master gewählt werden. Sie umfasst die Seminare, Praktika, Bachelor- und Masterarbeiten und die Wahlpflichtveranstaltungen.

Dabei sind die Module zu gleichen Teilen auf die beiden Lehrgebiete Datenbanken (DB) und Software Engineering (SWE) verteilt. Kernstück ist ein die beiden Lehrgebiete übergreifendes Praktikum (genannt ISE-Projekt) im Umfang von 16 LP, in dem im Team für externe Kunden ein Informationssystem mit ingenieurmäßigen Methoden unter Nutzung modernster Technologie entwickelt wird. Das ISE-Projekt wird immer im Wintersemester angeboten.

Vertiefende Module aus dem Gebiet SWE (aus dem Modulhandbuch des Masters)

ISWRE (8LP): Requirements Engineering

ISWArch (8LP): Komponenten und service-orientierte Systeme

ISWQM (8LP): Qualitätsmanagement

Vertiefende Module aus dem Gebiet DB (aus dem Modulhandbuch des Masters)

IDW (4LP) DataWarehouses

IRDB (8LP) Räumliche Datenbanken

IVDB (8 LP) Verteilte Datenbanken und Informationssysteme

IKDD (8LP) Knowledge Discovery in Databases

Weitere angebotene Module können auf Antrag hinzugenommen werden

DB: Management und Analyse von Datenströmen (4LP)

SWE: IT-Projektmanagement (3LP)

SWE: Software-Ökonomie (3LP)

SWE: Software-Evolution (3LP)

SWE: Wissensmanagement in der Softwareentwicklung (3LP)

| Veranstaltung                           | Semester | LP  |
|---|----------|-----|
| Wiss. Arbeiten                          | 1        | 2   |
| Seminar – Lehrgebiet SWE                | 1-3      | 4   |
| Seminar – Lehrgebiet DB                 | 1-3      | 4   |
| ISE Projekt                             | 1-3      | 16  |
| 2 Vertiefende Module Lehrgebiet SWE     | 1-3      | 16  |
| 2 Vertiefende Vorlesungen Lehrgebiet DB | 1-3      | 16  |
| Weitere Module                          | 1-3      | 6   |
| Effiziente Algorithmen 1                | 1-3      | 8   |
| Masterarbeit SWE oder DB                | 4        | 30  |
| Anwendungsgebiet                        | 1-3      | 18  |
| LP Summe                                |          | 120 |

Wurde das ISE-Projekt schon im Bachelor gemacht, so sollten im Master je 1 Praktikum in DB und SWE gemacht werden. Bei Belegung der Vertiefung im Bachelor müssen die im Master belegten Module disjunkt zu denen im Bachelor sein.

### ***Vertiefung Optimierung***

Diese Vertiefung hat als Inhalt Entwurf, Analyse und Implementierung von Algorithmen zur Lösung von kontinuierlichen und ganzzahligen Optimierungsproblemen. Sie kann im Bachelor- oder Master-Studium sowie durchgängig vom Bachelor- zum Master-Studium gewählt werden. Sie umfasst neben Vorlesungen, Seminaren und Praktika auch die Bachelor- bzw Masterarbeit. Empfohlene Voraussetzungen sind gute bis sehr gute Kenntnisse aus den Pflichtmodulen Mathematik.

Den Kern der Veranstaltungen dieser Vertiefung bilden die folgenden Module aus dem Gebiet Optimierung

Effiziente Algorithmen 1 (IEA1)

Effiziente Algorithmen 2 (IEA2)

Algorithmische Optimierung 1 (MH16)

Algorithmische Optimierung 2 (MH17)

sowie aus dem Gebiet Wissenschaftliches Rechnen

Numerische Optimierung bei Differentialgleichungen (MH8)

Alle diese Module können dabei unabhängig voneinander absolviert werden. Wird die Vertiefung sowohl im Bachelor- als auch im Masterstudium gewählt, sollten alle 5 Module absolviert werden.

Weitere erforderliche Module sind

mindestens ein Seminar (4 LP),

mindestens ein Fortgeschrittenen-Praktikum (8 LP).

Empfohlen werden, je nach aktuellem Angebot, weitere Module wie z.B. Kompaktkurs Gemischt-ganzzahlige und kombinatorische Optimierung (MIP) und Modul Numerik (MD1) (besonders für Optimierung bei Differentialgleichungen). Möglich sind weiterhin

Lineare Optimierung (MD3)

Nichtlineare Optimierung (MD4)

Numerische Optimierung bei Differentialgleichungen (MH8)

Numerische Methoden der Kontinuumsmechanik (MH9)

Spezielle Themen der Numerik (MH10)

Spezielle Themen der Optimierung (MH11)

Mixed-Integer Programming and Combinatorial Optimization (IMIP)

| Veranstaltung                                    | Semester | LP  |
|--|----------|-----|
| Wissenschaftliches Arbeiten                      | 1        | 2   |
| Seminar Optimierung                              | 1-3      | 4   |
| Seminar (muss nicht Optimierung sein)            |          | 4   |
| Fortgeschrittenen-Praktikum Optimierung          | 1-3      | 8   |
| 3 aus den oben genannten 5 Modulen               | 1-3      | 24  |
| Weitere Module aus dem Gebiet Optimierung        | 1-3      | 8   |
| Weitere Module aus mindestens 2 anderen Gebieten | 1-3      | 22  |
| Masterarbeit Optimierung                         | 4        | 30  |
| Anwendungsgebiet                                 | 1-3      | 18  |
| LP Summe   |          | 120 |

Für weitere Module werden besonders die Gebiete Wissenschaftliches Rechnen, Bildverarbeitung und Datenbanken empfohlen. Bei Belegung der Vertiefung im Bachelor müssen die im Master belegten Module disjunkt zu denen im Bachelor sein.

## **Vertiefung Theoretische Informatik**

Die Vertiefung kann im Master gewählt werden. Sie umfasst Seminare, Wahlpflichtveranstaltungen und die Masterarbeit. Es sind 2 Seminare und 3 vertiefende Vorlesungen aus dem Gebiet der Theoretischen Informatik zu wählen und die Masterarbeit in diesem Gebiet anzufertigen. Weiter ist die Vorlesung Effiziente Algorithmen 1 verpflichtend. Weitere Lehrveranstaltungen in der Informatik sowie das Anwendungsgebiet können gemäß der Prüfungsordnung frei gewählt werden.

Vertiefende Vorlesungen aus dem Bereich Theoretische Informatik (aus dem Modulhandbuch des Masters)

MH14 (8LP): Berechenbarkeit und Komplexität 1

MH15 (6LP): Berechenbarkeit und Komplexität 2

IFSA (8LP): Formale Sprachen und Automatentheorie

Auf Antrag kann eine dieser Vorlesungen durch eine vertiefende Vorlesung über effiziente Algorithmen ersetzt werden.

Ein Beratungsgespräch im 1. Semester mit einem der Dozenten der Theoretischen Informatik zur Ausgestaltung der Vertiefung wird empfohlen.

| Veranstaltung  | Semester | LP  |
|--|----------|-----|
| Wiss. Arbeiten   | 1        | 2   |
| 2 Seminare – Lehrgebiet Theoretische Informatik                | 1-3      | 8   |
| 3 Vertiefende Vorlesungen - Lehrgebiet Theoretische Informatik | 1-3      | 22  |
| Effiziente Algorithmen 1                                       | 1-3      | 8   |
| Weitere Module Informatik (Wahlpflichtmodule)                  | 1-3      | 32  |
| Anwendungsgebiet   | 1-3      | 18  |
| Masterarbeit – Lehrgebiet Theoretische Informatik              | 4        | 30  |
| LP Summe   |          | 120 |

## **Vertiefung Wissenschaftliches Rechnen**

Die Vertiefung kann im Bachelor, im Master und durchgängig vom Bachelor zum Master gewählt werden. Sie umfasst die Seminare, Praktika, Bachelor- und Masterarbeiten und die Wahlpflichtveranstaltungen.

Vertiefende Vorlesungen aus dem Gebiet Wissenschaftliches Rechnen (aus dem Modulhandbuch des Masters)

|  |
|--|
| Numerik von Transportprozessen in porösen Medien (INTPM)               |
| Objektorientiertes Programmieren im Wissenschaftlichen Rechnen (IOPWR) |
| Paralleles Höchstleistungsrechnen (IPHR)                               |
| Parallele Lösung großer Gleichungssysteme (IPLGG)                      |
| Simulationswerkzeuge (ISIMW)   |
| Numerik (MD1)  |
| Wissenschaftliches Rechnen (MD5)                                       |
| Numerische Lineare Algebra (MH5)                                       |

|  |
|--|
| Numerik gewöhnlicher Differentialgleichungen (MH6) |
| Numerik partieller Differentialgleichungen (MH7)   |

Falls die Vertiefung noch nicht im Bachelor gewählt wurde, sollte der Modul MD1 gewählt werden

| Veranstaltung  | Semester | LP  |
|--|----------|-----|
| Wiss. Arbeiten   | 1        | 2   |
| 2 Seminare Wissenschaftliches Rechnen                                | 1-3      | 8   |
| Objektorientiertes Programmieren im Wissenschaftlichen Rechnen IOPWR | 1-3      | 6   |
| Numerik partieller Differentialgleichungen (MH7)                     | 1-3      | 8   |
| Numerische Optimierung bei Differentialgleichungen (MH8)             | 1-3      | 8   |
| Paralleles Höchstleistungsrechnen (IPHR)                             | 1-3      | 8   |
| Algorithmische Optimierung I (MH16)                                  | 1-3      | 8   |
| Eine weitere Veranstaltung aus: MH17, IPLGG, ISIMW, INTPM, IMIP      | 1-3      | 8   |
| Weitere Module aus mindestens 1 anderem Gebiet (empfohlen: SWE )     | 1-3      | 16  |
| Anwendungsgebiet   | 1-3      | 18  |
| Masterarbeit Wissenschaftliches Rechnen                              |          | 30  |
| LP Summe   |          | 120 |

Bei Belegung der Vertiefung im Bachelor müssen die im Master belegten Module disjunkt zu denen im Bachelor sein.

## Module aus der Informatik

|                                       |  |
|---------------------------------------|--|
| Modulbezeichnung:                     | <b>Algorithmische Geometrie (IAGM)</b>   |
| ggf. Modulniveau:                     |  |
| ggf. Kürzel:                          | IAGM   |
| ggf. Lehrveranstaltungen:             |  |
| Studiensemester:                      | Ab 3. Semester Bachelor, ab 1. Semester Master   |
| Modulverantwortliche(r):              | Studiendekan/ Studiendekanin der Informatik  |
| Dozent(in):                           | Prof. Dr. Heike Leite  |
| Sprache:                              | Deutsch  |
| Zuordnung zum Curriculum:             |  |
| Lehrform/ SWS:                        | Vorlesung 2 SWS, 1 SWS Übung   |
| Arbeitsaufwand:                       | 120 h<br>45 h Präsenzstudium<br>15 h Prüfungsvorbereitung<br>60 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)   |
| Kreditpunkte:                         | 4 LP   |
| Voraussetzungen nach Prüfungsordnung: |  |
| Empfohlene Vorkenntnisse:             | Algorithmen und Datenstrukturen  |
| Lernergebnisse:                       | Die Studierenden kennen die grundlegenden Algorithmen und Datenstrukturen der geometrischen Datenverarbeitung.<br>verstehen Grundkonzepte wie konvexe Hülle, Polygontriangulierung und effektive Punktsuche und sind in der Lage diese algorithmisch umzusetzen.<br>beherrschen wesentliche Datenstrukturen zur effizienten Speicherung und Weiterverarbeitung der Daten.<br>können die Komplexität der verschiedenen Algorithmen berechnen. |
| Inhalt:                               | Basiskonzepte<br>Effiziente Punktsuche<br>Voronoidiagramme<br>Delaunaytriangulierung<br>Allgemeine Suchstrukturen  |
| Studien-/ Prüfungsleistung:           | Bestehen einer schriftlichen oder mündlichen Abschlussprüfung  |
| Medienformen:                         | Folien / Tafel / Skript / Lehrbuch   |
| Literatur:                            | <ul style="list-style-type: none"> <li>J. O'Rourke: Computational Geometry in C,</li> </ul>  |

|  |   |
|--|---|
|  | <p>Cambridge University Press, 1998.</p> <ul style="list-style-type: none"> <li>• H. Edelsbrunner: Geometry and Topology of Mesh Generation, Cambridge University Press, 2001.</li> <li>• Mark de Berg, Otfried Cheong, Marc van Kreveld, Mark Overmars: Computational Geometry - Algorithms and Applications, 3rd edition, Springer, 2008.</li> <li>• Aktuelle Fachveröffentlichungen</li> </ul> |
|--|---|

|                                       |  |
|---------------------------------------|--|
| Modulbezeichnung:                     | <b>Computergraphik 1 (ICG1)</b>  |
| ggf. Modulniveau:                     |  |
| ggf. Kürzel:                          | ICG1   |
| ggf. Lehrveranstaltungen:             |  |
| Studiensemester:                      | Ab 3. Semester Bachelor, ab 1. Semester Master   |
| Modulverantwortliche(r):              | Studiendekan/ Studiendekanin der Informatik  |
| Dozent(in):                           | Dr. Susanne Krömker  |
| Sprache:                              | Englisch oder Deutsch  |
| Zuordnung zum Curriculum:             |  |
| Lehrform/ SWS:                        | Vorlesung 2 SWS, Übung 2 SWS   |
| Arbeitsaufwand:                       | 180 h; davon 60 h Präsenzstudium, 100 h Aufgabenbearbeitung, 20 h Prüfungsvorbereitung   |
| Kreditpunkte:                         | 6 LP   |
| Voraussetzungen nach Prüfungsordnung: |  |
| Empfohlene Vorkenntnisse:             | Einführung in die Praktische Informatik (IPR), Programmierkurs (IPK)   |
| Lernergebnisse:                       | Die Studierenden sind in der Lage, einfache Graphikfenster mit Inhalt zu belegen, sind mit 2D-Rastergraphik vertraut und in der Lage, mit Objekten in Weltkoordinaten über Graphikbibliotheken (OpenGL) zu interagieren, können mit homogenen Koordinaten umgehen und Projektions- sowie Transformationsmatrizen bestimmen. Sie kennen die verschiedenen Bufferkonzepte zur schnellen Darstellung realistischer Effekte, sind mit Farbkonzepten und Darstellungsmodellen vertraut, können lokale Lichtmodelle aus optischen Prinzipien herleiten und wissen, wie man Texturen für unterschiedliche Effekte einsetzt. Sie kennen außerdem die physikalischen Grundlagen für globale Lichtmodelle. |
| Inhalt:                               | Grundlagen der Graphikprogrammierung   |

|                                |  |
|--------------------------------|--|
|                                | Koordinatensysteme<br>Projektionen, Transformationen<br>Zeichenalgorithmen<br>Bufferkonzepte (z-Buffer, Double-Buffer, Stereo-Buffer)<br>Shading und Lichtmodelle<br>Graphikbibliothek OpenGL<br>Direct Rendering (lokale Verfahren)<br>Texturen und Abbildungsverfahren<br>Globale Verfahren<br>Raytracing (Povray)<br>Volume Rendering |
| Studien-/<br>Prüfungsleistung: | Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung   |
| Medienformen:                  | Folien / Skript / Lehrbuch   |
| Literatur:                     | J. D. Foley, A. Van Dam, S. K. Feiner, J. F. Hughes und R. L. Phillips: Computer Graphics: Principles and Practice. Addison-Wesley, 1996<br>D. Shreiner, M. Woo, J. Neider und T. Davis: OpenGL programming guide: the official guide to learning OpenGL, version 1.4. OpenGL Architecture Review Board, Addison-Wesley, 2004            |

|                                       |  |
|---------------------------------------|--|
| Modulbezeichnung:                     | <b>Computergraphik 2 (ICG2)</b>  |
| ggf. Modulniveau:                     |  |
| ggf. Kürzel:                          | ICG2   |
| ggf. Lehrveranstaltungen:             |  |
| Studiensemester:                      | Ab 4. Semester Bachelor, ab 1. Semester Master   |
| Modulverantwortliche(r):              | Studiendekan/ Studiendekanin der Informatik  |
| Dozent(in):                           | Dr. Susanne Krömker  |
| Sprache:                              | Englisch oder Deutsch  |
| Zuordnung zum Curriculum:             |  |
| Lehrform/ SWS:                        | Vorlesung 2 SWS, Übung 2 SWS   |
| Arbeitsaufwand:                       | 180 h; davon 60 h Präsenzstudium, 100 h Aufgabenbearbeitung, 20 h Prüfungsvorbereitung   |
| Kreditpunkte:                         | 6 LP   |
| Voraussetzungen nach Prüfungsordnung: |  |
| Empfohlene Vorkenntnisse:             | Einführung in die Praktische Informatik (IPR), Programmierkurs (IPK), Computergraphik 1 (ICG1)   |
| Lernergebnisse:                       | Die Studierenden sind mit weiterführenden Lichtmodellen vertraut, können vertex- und fragmentbasierte Shader programmieren, kennen die Vor- und Nachteile verschiedener globaler photorealistischer Verfahren, |

|                             |  |
|-----------------------------|--|
|                             | können nichtphotorealistische Verfahren sinnvoll einsetzen und mit Splines als Linien und Flächen umgehen.   |
| Inhalt:                     | Shading und Bidirectional Reflectance Distribution Function (BRDF), C for graphics (Cg), Radiosity, Photon Mapping, Nichtphotorealismus (NPR), Splines   |
| Studien-/ Prüfungsleistung: | Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung   |
| Medienformen:               | Folien / Skript / Lehrbuch   |
| Literatur:                  | J. D. Foley, A. Van Dam, S. K. Feiner, J. F. Hughes und R. L. Phillips: Computer Graphics: Principles and Practice. Addison-Wesley, 1996<br>Randima Fernando, Mark J. Kilgard: The Cg Tutorial. Addison-Wesley, 2003 |

|                                       |   |
|---------------------------------------|---|
| Modulbezeichnung:                     | <b>Compilerbau (ICOM)</b>   |
| ggf. Modulniveau:                     |   |
| ggf. Kürzel:                          | ICOM  |
| ggf. Lehrveranstaltungen:             |   |
| Studiensemester:                      | Ab 3. Semester Bachelor, ab 1. Semester Master  |
| Modulverantwortliche (r):             | Studiendekan/ Studiendekanin der Informatik   |
| Dozent(in):                           | Prof. Dr. Gerhard Reinelt   |
| Sprache:                              | Deutsch   |
| Zuordnung zum Curriculum:             |   |
| Lehrform/ SWS:                        | Vorlesung 4 SWS, Übung 2 SWS  |
| Arbeitsaufwand:                       | 240 h<br>90 h Präsenzstudium<br>15 h Prüfungsvorbereitung<br>135 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen) |
| Kreditpunkte:                         | 8 LP  |
| Voraussetzungen nach Prüfungsordnung: |   |
| Empfohlene Vorkenntnisse:             | Einführung in die Praktische Informatik, Programmierkurs, Algorithmen und Datenstrukturen                                   |
| Lernergebnisse:                       | Die Studierenden kennen den prinzipiellen Aufbau eines Compilers haben die erforderlichen Grundlagen zu formalen            |

|                             |  |
|-----------------------------|--|
|                             | <p>Sprachen gelernt sind mit den Problematiken von lexikalischer, syntaktischer und semantischer Analyse vertraut kennen die Techniken zur Compilererstellung sind über Optimierungsmöglichkeiten informiert sind in der Lage, vorhandene Werkzeuge zum Compilerbau zu benutzen können die erworbenen Kenntnisse auf andere Anwendungsbereiche, wie z.B. Aufbereitung strukturierter Daten, übertragen</p> |
| Inhalt:                     | <p>Überblick über grundlegende Techniken<br/> Grundlagen der formale Sprachen<br/> Lexikalische Analyse<br/> Top-Down-Syntaxanalyse<br/> Bottom-Up-Syntaxanalyse<br/> Syntaxgesteuerte Übersetzung<br/> Semantische Analyse<br/> Die C-Maschine<br/> 3-Adress-Code<br/> Optimierung<br/> Datenfluss-Analyse</p>  |
| Studien-/ Prüfungsleistung: | <p>Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung</p>  |
| Medienformen:               | <p>Folien / Tafel / Skript / Lehrbuch</p>  |
| Literatur:                  | <p>z.B. Aho, A.V., Lam, M.S., Sethi, R., Ullman, J.D.:<br/> Compilers: Principles, Techniques and Tools,<br/> Pearson - Addison-Wesley, 2006</p>   |

|                           |  |
|---------------------------|--|
| Modulbezeichnung          | <b>Computerspiele (ICS)</b>  |
| ggf. Modulniveau:         |  |
| ggf. Kürzel:              | ICS  |
| ggf. Lehrveranstaltungen: |  |
| Studiensemester:          | Ab 3. Semester Bachelor, ab 1. Semester Master   |
| Modulverantwortliche(r):  | Studiendekan/ Studiendekanin der Informatik  |
| Dozent(in):               | Prof. Dr. Jürgen Hesser  |
| Sprache:                  | Deutsch/Englisch (richtet sich nach den teilnehmenden Studenten)   |
| Zuordnung zum Curriculum: |  |
| Lehrform/ SWS:            | Vorlesung 3 SWS, Übung 3 SWS   |
| Arbeitsaufwand:           | <p>240 h<br/> 75 h Präsenzstudium<br/> 15 h Prüfungsvorbereitung<br/> 150 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)</p> |
| Kreditpunkte:             | 8 LP   |

|                                       |   |
|---------------------------------------|---|
| Voraussetzungen nach Prüfungsordnung: |   |
| Empfohlene Vorkenntnisse:             | Einführung in die Praktische Informatik, Programmierkurs, Algorithmen und Datenstrukturen   |
| Lernergebnisse:                       | Die Studierenden lernen die Konzepte von der informatischen Sicht kennen, was Computerspiele ausmacht und welche Herausforderungen damit verbunden sind. Sie lernen, wie man gute und effiziente Architekturkonzepte dafür entwickelt sowie wie man typische Probleme aus Graphik, Kollisionserkennung, Animation/Physik, Pfadplanung/KI umsetzt. Zudem lernen sie in den Übungen, wie man konkrete Spiele entwickelt, so dass sie in der Lage sind, eigene Spieleengines zu realisieren. |
| Inhalt:                               | Überblick über die Einteilung von Computerspielen<br>Architektur von Game Engines<br>Vorstellung von OGRE als einer open-source Game Engine<br>Graphik und Computerspiele: ein Überblick<br>Kollisionserkennungstechniken<br>Animationstechniken und Physik bei Computerspielen mit Fokus auf der open source Bibliothek Bullet<br>Pfadplanung und KI   |
| Studien-/ Prüfungsleistung:           | Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen oder mündlichen Abschlussprüfung  |
| Medienformen:                         | Folien / Lehrbuch   |
| Literatur:                            | Gregory et al: Game Engine Architecture<br>Ericson: Real-Time Collision Detection<br>Eberly: Game Physics<br>Millington: Artificial Intelligence for Games  |

|                           |  |
|---------------------------|--|
| Modulbezeichnung:         | <b>Foundations of Cryptography and Network Security (ICNS)</b> |
| ggf. Modulniveau:         |  |
| ggf. Kürzel:              | ICNS   |
| ggf. Lehrveranstaltungen: |  |
| Studiensemester:          | Ab 4. Semester Bachelor, ab 1. Semester Master                 |
| Modulverantwortliche (r): | Studiendekan/ Studiendekanin der Informatik                    |
| Dozent(in):               | Dr. Sebastian Gajek, Jan Seedorf                               |

|                                       |   |
|---------------------------------------|---|
| Sprache:                              | Deutsch / Englisch  |
| Zuordnung zum Curriculum:             |   |
| Lehrform/ SWS:                        | Vorlesung 2 SWS mit Hausaufgaben  |
| Arbeitsaufwand:                       | 90 h<br>30 h Präsenzstudium<br>30 h Aufgabenbearbeitung (evtl. in Gruppen)<br>30 h Selbststudium und Prüfungsvorbereitung   |
| Kreditpunkte:                         | 3 LP  |
| Voraussetzungen nach Prüfungsordnung: |   |
| Empfohlene Vorkenntnisse:             | Theoretische Informatik, Betriebssysteme und Netzwerke  |
| Lernergebnisse:                       | <p>Die Studierenden beherrschen die theoretischen Grundlagen von Kryptographie und haben ein grundlegendes Verständnis von kryptographischen Verfahren erlangt. Die wichtigsten Algorithmen und Protokolle zur Verschlüsselung und digitaler Signatur, die eine Vielzahl von praktischen IT-Sicherheits-applikationen fundamentieren, sind ihnen im Detail bekannt. Sie sind mit den mathematischen Modellen vertraut, kryptographische Protokolle zu analysieren und formal auf ihre Sicherheit zu beweisen.</p> <p>Die Studierenden sind mit den gängigen Bedrohungen in heutigen Computernetzen vertraut. Sie sind in der Lage, die damit verbundenden Sicherheitsrisiken einzuschätzen und bewerten zu können. Ferner haben die Studierenden grundlegende Kenntnisse über gängige Sicherheitstechniken und Verfahren zum Schutz von Computernetzen.</p> <p>Die Studierenden können sowohl aktuelle Sicherheitsprobleme in der Praxis identifizieren als auch effiziente Schutzmaßnahmen erarbeiten.</p> |
| Inhalt:                               | <p>Im ersten Teil der Veranstaltung werden theoretische Grundlagen der Kryptographie (z.B. one-way functions, permutation, pseudo-random generators, symmetrische Verschlüsselung, asymmetrische Verschlüsselung, digitale Signatur) und gängige kryptographische Verfahren und Protokolle (z.B. RSA, AES, SSL, IPSec, SSH) behandelt. Aufbauend auf diesen Grundlagen werden im zweiten Teil der Vorlesung praktische Anwendungen von kryptographischen Verfahren in Computernetzen sowie weitere Kernthemen im Bereich der Netzwerksicherheit behandelt. Themen sind unter anderem Zertifikate / Public Key</p>   |

|                             |  |
|-----------------------------|--|
|                             | Infrastructure (PKI), DNSSec, Kerberos, Intrusion Detection Systems, Firewalls und malizöse Software.  |
| Studien-/ Prüfungsleistung: | Erfolgreiche Abgabe von Hausaufgaben und Bestehen einer schriftlichen oder mündlichen Abschlussprüfung   |
| Medienformen:               | Folien / Tafel / Skript / Lehrbuch   |
| Literatur:                  | z.B. Dieter Gollmann: Computer Security, John Wiley & Sons; 2nd Edition (2005); William Stallings: Network Security Essentials, Pearson International, 2nd Edition |

|                                       |  |
|---------------------------------------|--|
| Modulbezeichnung:                     | <b>Data Mining - Algorithmen und parallele Verfahren (IDM)</b>   |
| ggf. Modulniveau:                     |  |
| ggf. Kürzel:                          | IDM  |
| ggf. Lehrveranstaltungen:             |  |
| Studiensemester:                      | Ab 5. Semester Bachelor, ab 1. Semester Master   |
| Modulverantwortliche(r):              | Studiendekan/ Studiendekanin der Informatik  |
| Dozent(in):                           | Prof. Dr. Artur Andrzejak  |
| Sprache:                              | Deutsch  |
| Zuordnung zum Curriculum:             |  |
| Lehrform/SWS:                         | Vorlesung 2 SWS, Übung 2 SWS mit Diskussion der Übungsaufgaben.  |
| Arbeitsaufwand:                       | 180 h  |
| Kreditpunkte:                         | 6 LP   |
| Voraussetzungen nach Prüfungsordnung: | Keine  |
| Empfohlene Vorkenntnisse:             | Programmierkenntnisse (z.B. in Java) und elementare Wahrscheinlichkeitsrechnung; Vorlesung „Knowledge Discovery in Databases (IKDD)“ ist empfohlen aber nicht notwendig  |
| Lernergebnisse:                       | Die Studierenden<br>verstehen die Nützlichkeit und Anwendungsgebiete einzelner Verfahren des Data Mining<br>verstehen die Ansätze der parallelen Programmierung und der parallelen Datenanalyse, insbesondere Map-Reduce und GPU-Programmierung<br>kennen die Methoden und Algorithmen der Vorverarbeitung von Daten, insbesondere parallele Attributselektion und Dimensionsreduktion |

|                             |  |
|-----------------------------|--|
|                             | <p>(PCA)</p> <p>haben grundlegendes Verständnis der Klassifikation, Regression und Clustering sowie kennen ausgewählte parallele Implementierungen dieser Verfahren</p> <p>sind in der Lage, die Ergebnisse zu evaluieren und haben Verständnis der Problematik der Überanpassung (Overfitting)</p> <p>haben Fähigkeit der praktischen Verwendung von Data Mining Tools, Bibliotheken und Frameworks wie KNIME, WEKA, GNU Octave, NumPy/SciPy, Hadoop</p>  |
| Inhalt:                     | <p>In diesem Modul werden die Verfahren des Data Mining mit Fokus auf Parallelisierung, Programmierertools und ihre sinnvolle Verwendung vorgestellt. Zu den Inhalten zählen:</p> <ul style="list-style-type: none"> <li>Grundbegriffe und Übersicht der Data Mining Verfahren</li> <li>Einführung in Tools, Bibliotheken und Frameworks</li> <li>Grundlagen der parallelen Programmierung (Map-Reduce, GPUs, Actors) für die Datenanalyse</li> <li>Verfahren zur Vorverarbeitung der Daten inklusive Attributselektion und Dimensionsreduktion</li> <li>Klassifikation, Regression, Clustering</li> <li>Bayes'sche Netze</li> <li>Evaluation der Ergebnisse, Overfitting und ihre Vermeidung</li> </ul> |
| Studien-/ Prüfungsleistung: | Erfolgreiche Teilnahme an den Übungen (Erreichen einer Mindestpunktzahl) und Bestehen einer Abschlussprüfung   |
| Medienformen:               | Folien/ Tafel/ E-Learning/ Lehrbuch  |
| Literatur:                  | <p>z.B.</p> <p>Ethem Alpaydin, <b>Maschinelles Lernen</b>, Oldenbourg Verlag, 2008</p> <p>Trevor Hastie, Robert Tibshirani, Jerome Friedman, <b>The Elements of Statistical Learning: Data Mining, Inference, and Prediction</b>, Springer, 2009 (<a href="#">online</a>)</p> <p>Stephen Marsland, <b>Machine Learning: An Algorithmic Perspective</b>, CRC Press Inc., 2009</p> <p>Ron Bekkerman, Misha Bilenko, John Langford, <b>Scaling Up Machine Learning</b>, Cambridge University Press, 2012</p>  |

|                   |                              |
|-------------------|------------------------------|
| Modulbezeichnung: | <b>Data Warehouses (IDW)</b> |
|-------------------|------------------------------|

|                                       |  |
|---------------------------------------|--|
| ggf. Modulniveau:                     |  |
| ggf. Kürzel:                          | IDW  |
| ggf. Lehrveranstaltungen:             |  |
| Studiensemester:                      | Ab 5. Semester Bachelor, ab 1. Semester Master   |
| Modulverantwortliche(r):              | Studiendekan/Studiendekanin der Informatik   |
| Dozent(in):                           | Prof. Dr. Michael Gertz  |
| Sprache:                              | Deutsch/Englisch nach Vereinbarung   |
| Zuordnung zum Curriculum:             |  |
| Lehrform/ SWS:                        | Vorlesung 2 SWS, Übungen 1 SWS   |
| Arbeitsaufwand:                       | 120 h; davon<br>30 h Präsenzstudium<br>15 h Prüfungsvorbereitung<br>75 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)  |
| Kreditpunkte:                         | 4 LP   |
| Voraussetzungen nach Prüfungsordnung: | Keine  |
| Empfohlene Vorkenntnisse:             | Algorithmen und Datenstrukturen (IAD), Datenbanken 1 (DB1)   |
| Lernergebnisse:                       | Die Studierenden sind in der Lage, zu einer gegebenen Anwendung eine Data Warehouse Pipeline zu modellieren kennen die Komponenten der Architektur von Data Warehouse Systemen wissen, welche Arten von Indexe und Speicherstrukturen typische Data Warehouse Anfragen unterstützen kennen Verfahren der Anfragebearbeitung und (manueller) Optimierungstechniken wissen, wie materialisierte Sichten einzusetzen sind und wie diese zu pflegen sind sind mit dem Aufsetzen eines realen Data Warehouses und den administrativen Tasks vertraut kennen die wichtigsten Performance Benchmarks für Data Warehouse Anwendungen |
| Inhalt:                               | Einführung & Grundbegriffe<br>Data-Warehouse-Architekturen<br>Modellierung von Data Warehouses<br>Index- und Speicherungsstrukturen<br>Anfragen an Data Warehouses<br>Anfrageverarbeitung und -optimierung<br>Materialisierte Sichten  |
| Studien-/ Prüfungsleistung:           | Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und erfolgreiche Teilnahme an einer schriftlichen Prüfung   |

|               |  |
|---------------|--|
| Medienformen: | Folien/ Tafel/ E-Learning/ Lehrbuch  |
| Literatur:    | z. B.: Wolfgang. Lehner. Datenbanktechnologie für Data-Warehouse-Systeme. dpunkt.verlag, Heidelberg, 2003.<br>Andreas Bauer, Holger Günzel. Data Warehouse Systeme – Architektur, Entwicklung, Anwendung. 3. Auflage, dpunkt.verlag, Heidelberg, 2008.<br>Gunter Saake, Andreas Heuer, Kai-Uwe Sattler. Datenbanken: Implementierungstechniken. 2. Auflage, mitp-Verlag, Bonn, 2005. |

|                                       |   |
|---------------------------------------|---|
| Modulbezeichnung:                     | <b>Effiziente Algorithmen 1 (IEA1)</b>  |
| ggf. Modulniveau:                     |   |
| ggf. Kürzel:                          | IEA1  |
| ggf. Lehrveranstaltungen:             |   |
| Studiensemester:                      | Ab 3. Semester Bachelor, ab 1. Semester Master  |
| Modulverantwortliche (r):             | Studiendekan/ Studiendekanin der Informatik   |
| Dozent(in):                           | Prof. Dr. Gerhard Reinelt   |
| Sprache:                              | Deutsch   |
| Zuordnung zum Curriculum:             |   |
| Lehrform/ SWS:                        | Vorlesung 4 SWS, Übungen 2 SWS  |
| Arbeitsaufwand:                       | 240 h<br>90 h Präsenzstudium<br>15 h Prüfungsvorbereitung<br>135 h Selbststudium und Aufgabebearbeitung (evtl. in Gruppen)  |
| Kreditpunkte:                         | 8 LP  |
| Voraussetzungen nach Prüfungsordnung: |   |
| Empfohlene Vorkenntnisse:             | Einführung in die Praktische Informatik, Programmierkurs, Algorithmen und Datenstrukturen   |
| Lernergebnisse:                       | Die Studierenden verstehen die grundlegenden Konzepte der Graphentheorie<br>können Fragestellungen als kombinatorische Optimierungsprobleme modellieren<br>können die Komplexität von Optimierungsproblemen analysieren<br>kennen die Methoden zum Beweis der Korrektheit kombinatorischer Algorithmen und der Analyse ihrer Laufzeit<br>kennen die grundlegenden algorithmischen Ansätze |

|                             |   |
|-----------------------------|---|
|                             | sind mit den Fragen der effizienten Implementierung vertraut<br>haben Einblick in die vielfältigen Anwendungsgebiete der kombinatorischen Optimierung   |
| Inhalt:                     | Grundbegriffe der Graphentheorie<br>Grundlegende Graphenalgorithmien<br>Optimale Bäume und Branchings<br>Kürzeste Wege<br>Das Zuordnungsproblem<br>Maximale Flüsse<br>Minimale Schnitte in ungerichteten Graphen<br>Flüsse mit minimalen Kosten<br>Matchingprobleme |
| Studien-/ Prüfungsleistung: | Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung  |
| Medienformen:               | Folien / Tafel / Skript / Lehrbuch  |
| Literatur:                  | z.B. Korte, B., Vygen, J.: Kombinatorische Optimierung, Springer, 2007<br>Cook, W.J., Cunningham, W.H., Pulleyblank, W.R., Schrijver, A.: Combinatorial Optimization, Wiley, 1997   |

|                           |   |
|---------------------------|---|
| Modulbezeichnung:         | <b>Effiziente Algorithmen 2 (IEA2)</b>  |
| ggf. Modulniveau:         |   |
| ggf. Kürzel:              | IEA2  |
| ggf. Lehrveranstaltungen: |   |
| Studiensemester:          | Ab 3. Semester Bachelor, ab 1. Semester Master  |
| Modulverantwortliche (r): | Studiendekan/ Studiendekanin der Informatik   |
| Dozent(in):               | Prof. Dr. Gerhard Reinelt   |
| Sprache:                  | Deutsch   |
| Zuordnung zum Curriculum: |   |
| Lehrform/ SWS:            | Vorlesung 4 SWS, Übungen 2 SWS  |
| Arbeitsaufwand:           | 240 h<br>90 h Präsenzstudium<br>15 h Prüfungsvorbereitung<br>135 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen) |
| Kreditpunkte:             | 8 LP  |
| Voraussetzungen nach      |   |

|                             |   |
|-----------------------------|---|
| Prüfungsordnung:            |   |
| Empfohlene Vorkenntnisse:   | Einführung in die Praktische Informatik, Programmierkurs, Algorithmen und Datenstrukturen, Absolvierung des Moduls IEA1 ist nützlich, aber nicht Voraussetzung  |
| Lernergebnisse:             | Die Studierenden können polynomiale und NP-schwere Optimierungsprobleme klassifizieren kennen die gesamte Bandbreite von Lösungsansätzen für schwierige Probleme können geeignete Modellierungen für Anwendungsprobleme selbst erstellen sind in der Lage, geeignete Lösungsverfahren auszuwählen können selbst Lösungsverfahren konzipieren und implementieren   |
| Inhalt:                     | NP-schwere Optimierungsprobleme approximative Algorithmen und Heuristiken (Bin-Packing, Scheduling, Knapsack, Traveling Salesman) Relaxierungen (lineare, kombinatorische, semidefinite, Lagrange-Relaxierungen) Verfahren zur Bestimmung optimaler Lösungen (dynamische Optimierung, Branch-and-Bound, Branch-and-Cut) lineare 0/1- Optimierung (Modellierung, Schnittebenen) polyedrische Kombinatorik, Spaltengenerierung und Dekomposition Fallstudien (Traveling-Salesman_Problem, Max-Cut-Problem.) Es wird somit das gesamte Spektrum von Lösungsverfahren für schwierige kombinatorische Probleme vermittelt. |
| Studien-/ Prüfungsleistung: | Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung  |
| Medienformen:               | Folien / Tafel / Skript / Lehrbuch  |
| Literatur:                  | z.B. Korte, B., Vygen, J.: Kombinatorische Optimierung, Springer, 2007<br>Nemhauser, G.L., Wolsey, L.A.: Integer Combinatorial Optimization, Wiley, 1988  |

|                   |  |
|-------------------|--|
| Modulbezeichnung: | <b>Software-Praktikum für Fortgeschrittene (IFM)</b> |
| ggf. Modulniveau: |  |
| ggf. Kürzel:      | IFM  |

|  |  |
|--|--|
| ggf.<br>Lehrveranstaltungen:             |  |
| Studiensemester:                         | Ab 3. Semester Bachelor, ab 1. Semester Master   |
| Modulverantwortliche(r):                 | Studiendekan/ Studiendekanin   |
| Dozent(in):                              | Dozenten/Dozentinnen der Informatik  |
| Sprache:                                 | Deutsch  |
| Zuordnung zum<br>Curriculum:             |  |
| Lehrform/ SWS:                           | Praktikum 6 SWS  |
| Arbeitsaufwand:                          | 240 h; davon<br>mind. 25 h Präsenzstunden<br>10 h Vorbereitung Vortrag   |
| Kreditpunkte:                            | 8 LP   |
| Voraussetzungen nach<br>Prüfungsordnung: |  |
| Empfohlene<br>Vorkenntnisse:             |  |
| Lernergebnisse:                          | <p>Die Studierenden erlangen vertiefende Problemlösungskompetenz für komplexe Entwurfs- und Implementierungsaufgaben können Problemanalyse- und Beschreibungstechniken klar darstellen, differenzieren und anwenden vertiefen Programmierkenntnisse in der jeweiligen für das Projekt erforderlichen Programmiersprache sind in der Lage, das Projekt mit Hilfe einer Softwareentwicklungsumgebung durchzuführen</p> <p>Zusätzlich werden die projektypischen Kompetenzen vertieft, insbesondere das Arbeiten im Team (von bis zu drei Studierenden): Durchführung und Evaluation von Projekten und ihrer Phasenstruktur Planung und Durchführung von Projekt- und Teamarbeit.</p> <p>Zu den zu trainierenden Softskills zählen somit insbesondere Teamfähigkeit, Verfeinerung von Präsentationstechniken, etwaige Erschließung wissenschaftlicher Literatur sowie eigenverantwortliches Arbeiten.</p> |
| Inhalt:                                  | <p>Domänenkenntnisse abhängig von den DozentInnen; allgemeine Lerninhalte sind: Vertiefung in die Projektarbeit Eigenständige Entwicklung von komplexer Software und deren Dokumentation</p>   |
| Studien-/<br>Prüfungsleistung:           | Bewertung der dokumentierte Software, des Projektberichts und des Vortrag  |

|               |                                     |
|---------------|-------------------------------------|
| Medienformen: | Folien/ Tafel/ E-Learning/ Lehrbuch |
| Literatur:    |                                     |

|                                       |  |
|---------------------------------------|--|
| Modulbezeichnung:                     | <b>Formale Sprachen und Automatentheorie (IFSA)</b>  |
| ggf. Modulniveau:                     |  |
| ggf. Kürzel:                          | IFSA   |
| ggf. Lehrveranstaltungen:             |  |
| Studiensemester:                      | Ab 3. Semester Bachelor, ab 1. Semester Master   |
| Modulverantwortliche(r):              | Studiendekan/ Studiendekanin der Informatik  |
| Dozent(in):                           | Prof. Dr. Klaus Ambos-Spies, Priv.-Doz. Dr. Wolfgang Merkle  |
| Sprache:                              | Deutsch  |
| Zuordnung zum Curriculum:             |  |
| Lehrform/ SWS:                        | Vorlesung 4 SWS, Übungen 2 SWS mit Hausaufgaben.   |
| Arbeitsaufwand:                       | 240 h  |
| Kreditpunkte:                         | 8 LP   |
| Voraussetzungen nach Prüfungsordnung: |  |
| Empfohlene Vorkenntnisse:             | ITH  |
| Lernergebnisse:                       | <p>Die Studierenden kennen die Stufen der Chomsky-Hierarchie und deren äquivalente Beschreibung durch Grammatiken und Automatenmodelle, kennen die Charakterisierungen der Klasse der regulären Sprachen durch rechts- bzw. linkslineare Grammatiken, durch deterministische und nichtdeterministische endliche Automaten, durch reguläre Ausdrücke und durch Rechtskongruenzen mit endlichem Index, sind in der Lage, das Pumping-Lemma für reguläre Sprachen zu beweisen und dieses anzuwenden, um zu zeigen, dass bestimmte Sprachen nicht regulär sind, kennen den Nachweis der Eindeutigkeit des Minimalautomaten und können diesen in Form eines reduzierten Automaten oder eines Äquivalenzklassenautomaten konstruieren, können das Pumping-Lemma für kontextfreie Sprachen beweisen und dieses anwenden, um zu zeigen, dass bestimmte Sprachen nicht kontextfrei sind, können die Klasse der kontextfreien Sprachen</p> |

|                             |  |
|-----------------------------|--|
|                             | durch kontextfreie Grammatiken und durch Kellerautomaten beschreiben und die Äquivalenz dieser Darstellungen beweisen, kennen den Satz von Chomsky und Schützenberger und seine anschauliche Bedeutung für die Struktur der Ableitungen in kontextfreien Sprachen, sind mit Normalformen, Abschlusseigenschaften, Entscheidbarkeits- und Komplexitätsaspekten insbesondere der regulären und kontextfreien Sprachen vertraut, können die Klasse der kontextsensitiven Sprachen durch kontextsensitive Grammatiken und Grammatiken vom Erweiterungstyp sowie durch linear beschränkte Automaten beschreiben und die Äquivalenz dieser Darstellungen beweisen. |
| Inhalt:                     | Die Vorlesung gibt eine Einführung in die Theorie der formalen Sprachen und die dort verwendeten Begriffe und Methoden zur syntaktischen Sprachbeschreibung und -analyse unter besonderer Berücksichtigung algorithmischer Aspekte. Der Schwerpunkt der Vorlesung liegt auf den Sprachklassen der Chomsky-Hierarchie und deren Charakterisierung hinsichtlich der Beschreibbarkeit durch Grammatiken und der Erkennbarkeit durch Automaten.  |
| Studien-/ Prüfungsleistung: | Erfolgreiche Teilnahme an den Übungen (Erreichen einer Mindestpunktzahl) und Bestehen einer mündlichen Abschlussprüfung.   |
| Medienformen:               | Folien/ Tafel/ E-Learning/ Lehrbuch  |
| Literatur:                  |  |

|                           |  |
|---------------------------|--|
| Modulbezeichnung:         | <b>Flow and Transport in Terrestrial Systems (IFTTS)</b>         |
| ggf. Modulniveau:         |  |
| ggf. Kürzel:              | IFTTS  |
| ggf. Lehrveranstaltungen: |  |
| Studiensemester:          | Ab 3. Semester Bachelor, ab 1. Semester Master                   |
| Modulverantwortliche (r): | Studiendekan/ Studiendekanin der Informatik                      |
| Dozent(in):               | Dr. Olaf Ippisch, Prof. Dr. Kurt Roth, Prof. Dr. Hans-Jörg Vogel |
| Sprache:                  | Englisch   |
| Zuordnung zum Curriculum: | Wird unregelmäßig angeboten                                      |
| Lehrform/ SWS:            | Blockvorlesung mit Übung 5tägig (2 SWS)                          |

|                                       |   |
|---------------------------------------|---|
| Arbeitsaufwand:                       | 90 h<br>40 h Präsenzstudium<br>50 h Selbststudium und Nachbereitung   |
| Kreditpunkte:                         | 3 LP  |
| Voraussetzungen nach Prüfungsordnung: |   |
| Empfohlene Vorkenntnisse:             | Grundkenntnisse über terrestrische Systeme (MVEnv2 oder INTPM) oder die Numerik partieller Differentialgleichungen (MH7).   |
| Lernergebnisse:                       | Die Studierenden<br>haben die in natürlichen terrestrischen Systemen ablaufenden Transportprozesse gründlich verstanden.<br>kennen die Grenzen der üblichen Modelle dieser Prozesse.<br>können moderne numerische Simulationswerkzeuge effektiv einsetzen.  |
| Inhalt:                               | Fundamentale physikalische Prozesse und aktuelle Konzepte zum Transport von Wasser und gelösten Stoffen im Untergrund<br>Mehrskalige Architektur sedimentärer Formationen<br>Makroskopische Phänomene und effektive Beschreibungen<br>Numerische Simulation von Fluss und Transport in poröser Medien<br>Wechselwirkung Boden-Atmosphäre. |
| Studien-/ Prüfungsleistung:           | Mündliche oder schriftliche Abschlussprüfung (je nach Teilnehmerzahl)   |
| Medienformen:                         | Folien / Tafel / Skript   |
| Literatur:                            |   |

|                           |  |
|---------------------------|--|
| Modulbezeichnung:         | <b>Grundlagen der wissenschaftlichen Visualisierung (IGVV)</b> |
| ggf. Modulniveau:         |  |
| ggf. Kürzel:              | IGVV   |
| ggf. Lehrveranstaltungen: |  |
| Studiensemester:          | Ab 3. Semester Bachelor, ab 1. Semester Master                 |
| Modulverantwortliche(r):  | Studiendekan/ Studiendekanin der Informatik                    |
| Dozent(in):               | Prof. Dr. Heike Leite  |
| Sprache:                  | Deutsch  |
| Zuordnung zum Curriculum: |  |

|                                       |   |
|---------------------------------------|---|
| Lehrform/ SWS:                        | Vorlesung 4 SWS, Übung 2 SWS  |
| Arbeitsaufwand:                       | 240 h<br>90 h Präsenzstudium<br>15 h Prüfungsvorbereitung<br>135 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)   |
| Kreditpunkte:                         | 8 LP  |
| Voraussetzungen nach Prüfungsordnung: |   |
| Empfohlene Vorkenntnisse:             | Einführung in die Praktische Informatik, Programmierkurs, Algorithmen und Datenstrukturen   |
| Lernergebnisse:                       | Die Studierenden verstehen wie die menschliche visuelle Wahrnehmung abläuft und worauf man bei der visuellen Darstellung von Information aus wahrnehmungspsychologischer Sicht achten muss.<br>kennen die Verarbeitungskette in der Visualisierung.<br>kennen die grundlegenden Algorithmen zur Visualisierung von räumlichen und nicht-räumlichen Skalar- und Vektordaten.<br>verstehen die Designentscheidungen bei der Implementierung von interaktiven System zur Datenexploration und können diese praktisch umsetzen.<br>können selbständig Visualisierungsalgorithmen und GUI-basierte Anwendungsprogramme implementieren. |
| Inhalt:                               | Geschichte der Visualisierung<br>Menschliche Wahrnehmung<br>Verarbeitungskette in der Visualisierung<br>Visualisierung von Skalar- und Vektordaten<br>Visualisierung von Graphen<br>Interaktion und Datenexploration  |
| Studien-/ Prüfungsleistung:           | Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen oder mündlichen Abschlussprüfung  |
| Medienformen:                         | Folien / Tafel / Skript / Lehrbuch  |
| Literatur:                            | <ul style="list-style-type: none"> <li>• Alexandru C. Telea: Data Visualization - Principles and Practice, A K Peters Ltd., 2007.</li> <li>• Robert Spence: Information Visualization, Addison Wesley, 2000.</li> <li>• Colin Ware: Information Visualization, Morgan Kaufmann, 2. Edition, 2004.</li> </ul>  |

|                                       |   |
|---------------------------------------|---|
| Modulbezeichnung:                     | <b>Inverse Problems (IIP)</b>   |
| ggf. Modulniveau:                     |   |
| ggf. Kürzel:                          | IIP   |
| ggf. Lehrveranstaltungen:             |   |
| Studiensemester:                      | Ab 3. Semester Bachelor, ab 1. Semester Master  |
| Modulverantwortliche (r):             | Studiendekan/ Studiendekanin der Informatik   |
| Dozent(in):                           | Prof. Dr. Jürgen Hesser   |
| Sprache:                              | Englisch  |
| Zuordnung zum Curriculum:             | (scientific computing – wenn diese Vertiefung aktuell ist, sonst Wahlfach)  |
| Lehrform/ SWS:                        | Vorlesung 2 SWS, Übung 2 SWS und Hausarbeiten   |
| Arbeitsaufwand:                       | 240 h<br>60 h Präsenzstudium<br>15 h Prüfungsvorbereitung<br>165 h Selbststudium und Aufgabenbearbeitung, Hausarbeiten  |
| Kreditpunkte:                         | 8 LP  |
| Voraussetzungen nach Prüfungsordnung: |   |
| Empfohlene Vorkenntnisse:             | Einführung in die Praktische Informatik, Programmierkurs, Algorithmen und Datenstrukturen, Numerische Mathematik  |
| Lernergebnisse:                       | Die Studierenden verstehen, was inverse Probleme sind und warum sie schwer zu lösen sind. Sie lernen die Prinzipien kennen, wie man sowohl deterministische Probleme, als auch stochastische Probleme löst und dabei auch die Regularisierungsparameter geeignet wählt. Schließlich erfahren sie die neuesten Entwicklungen im Bereich compressed sensing. Alle Prinzipien werden an zwei ausgewählten Gebieten, der Tomographie und des Deblurrings dargestellt. Sie erhalten damit die Kompetenz komplexe Probleme zu lösen, die mit klassischen Techniken nicht stabil lösbar sind und sind damit in der Lage auch komplexe experimentelle Messungen adäquat auswerten zu können |
| Inhalt:                               | Deterministische inverse Probleme<br>Stochastische inverse Probleme<br>Wahl der Regularisierungsparameter<br>Compressed sensing<br>Tomographie<br>Deblurring  |

|                                |  |
|--------------------------------|--|
| Studien-/<br>Prüfungsleistung: | Erfolgreiche Teilnahme an den Übungen (mehr als 60% der Punkte müssen erreicht werden) und Bestehen einer mündlichen Abschlussprüfung  |
| Medienformen:                  | PDF-Folien, Lehrbuch, Matlab-Beispiele   |
| Literatur:                     | M. Bertero, P. Boccacci: Introduction to Inverse Problems in Imaging, IoP, 2002<br>web-Page and book:<br><a href="http://www.slaney.org/pct/pct-toc.html">http://www.slaney.org/pct/pct-toc.html</a> |

|  |  |
|--|--|
| Modulbezeichnung:                        | <b>Knowledge Discovery in Databases (IKDD)</b>   |
| ggf. Modulniveau:                        |  |
| ggf. Kürzel:                             | IKDD   |
| ggf.<br>Lehrveranstaltungen:             |  |
| Studiensemester:                         | Ab 5. Semester Bachelor, ab 1. Semester Master   |
| Modulverantwortliche(r):                 | Studiendekan/Studiendekanin der Informatik   |
| Dozent(in):                              | Prof. Dr. Michael Gertz  |
| Sprache:                                 | Deutsch/Englisch nach Vereinbarung   |
| Zuordnung zum<br>Curriculum:             |  |
| Lehrform/ SWS:                           | Vorlesung 4 SWS, Übungen 2 SWS   |
| Arbeitsaufwand:                          | 240 h; davon<br>90 h Präsenzstudium<br>15 h Prüfungsvorbereitung<br>135 h Selbststudium und Aufgabenbearbeitung<br>(eventuell in Gruppen)  |
| Kreditpunkte:                            | 8 LP   |
| Voraussetzungen nach<br>Prüfungsordnung: | Keine  |
| Empfohlene<br>Vorkenntnisse:             | Algorithmen und Datenstrukturen (IAD),<br>Datenbanken 1 (DB1)  |
| Lernergebnisse:                          | Die Studierenden<br>sind in der Lage, die Daten zu einem Data Mining<br>Verfahren mit Hilfe geeigneter (statistischer)<br>Methoden vorzuverarbeiten<br>kennen die grundlegenden Verfahren der<br>statistischen Datenanalyse<br>sind in der Lage, grundlegende Techniken des<br>Data Mining auf Daten aus verschiedenen<br>Anwendungsbereichen anzuwenden<br>sind in der Lage, Data Mining Techniken im<br>Kontext von Datenbanken und des KDD Prozesses<br>zu realisieren und anzuwenden<br>kennen die Techniken und Prinzipien, die den<br>Algorithmen zur Klassifikation und dem Clustering<br>von Daten zugrunde liegen |

|                             |  |
|-----------------------------|--|
|                             | <p>wissen, wie welche Clustering Algorithmen im Kontext welcher Daten und Anwendungen zu verwenden sind</p> <p>kennen die Verfahren zum Finden häufiger Muster in Daten.</p> <p>kennen die wichtigsten Techniken zur Erkennung von Ausreißern</p> <p>sind vertraut mit den grundlegenden Algorithmen und Techniken zur Analyse von zeit- und raumbezogenen Daten</p> |
| Inhalt:                     | <p>Grundbegriffe: Statistik und Daten</p> <p>Datenaufbereitung</p> <p>Clustering</p> <p>Frequent Pattern Mining</p> <p>Klassifikationsverfahren</p> <p>Mining von Graphen</p> <p>Mining von räumlich und zeitlich veränderlichen Daten (z.B. Objekt-Trajektorien und Zeitreihen)</p>   |
| Studien-/ Prüfungsleistung: | <p>Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und erfolgreiche Teilnahme an einer schriftlichen Prüfung</p>  |
| Medienformen:               | <p>Folien/ Tafel/ E-Learning/ Lehrbuch</p>   |
| Literatur:                  | <p>z. B.: Jiawei Han und Micheline Kamber: Data Mining. Concepts and Techniques, Morgan Kaufmann Series in Data Management Systems, 2006.</p> <p>Martin Ester und Jörg Sander: Knowledge Discovery in Databases: Techniken und Anwendungen, Springer, 2000.</p>  |

|                           |  |
|---------------------------|--|
| Modulbezeichnung:         | <b>Künstliche Intelligenz (IKI)</b>            |
| ggf. Modulniveau:         |  |
| ggf. Kürzel:              | IKI  |
| ggf. Lehrveranstaltungen: |  |
| Studiensemester:          | Ab 3. Semester Bachelor, ab 1. Semester Master |
| Modulverantwortliche(r):  | Studiendekan/Studiendekanin der Informatik     |
| Dozent(in):               | Prof. Dr. Björn Ommer                          |
| Sprache:                  | Deutsch oder Englisch                          |
| Zuordnung zum Curriculum: | Wird unregelmäßig angeboten                    |
| Lehrform/ SWS:            | Vorlesung 2 SWS, Übung 2 SWS                   |
| Arbeitsaufwand:           | 180 h<br>60h Präsenzstudium                    |

|   |   |
|---|---|
|   | 15h Prüfungsvorbereitung<br>105h Selbststudium und Bearbeitung von<br>theoretischen Aufgaben und Programmierübungen   |
| Kreditpunkte:                               | 6 LP  |
| Voraussetzungen<br>nach<br>Prüfungsordnung: |   |
| Empfohlene<br>Vorkenntnisse:                |   |
| Lernergebnisse:                             | Die Studierenden<br>haben gelernt Algorithmen zu entwickeln die<br>Probleme durch Suchen lösen<br>können logische Inferenz zur Lösungssuche<br>einsetzen<br>sind in der Lage Inferenz im Fall von unsicherem<br>Wissen über die Welt durchzuführen<br>können lernende Agenten entwickeln<br>können die erworbenen Kenntnisse auf<br>Anwendungsbereiche, wie z.B. Computersehen<br>oder Textmining, übertragen |
| Inhalt:                                     | Solving problems by searching<br>Game playing<br>Inference using logic<br>Knowledge bases<br>Planning and acting<br>Reasoning under uncertainty<br>Machine Learning<br>Applications   |
| Studien-/<br>Prüfungsleistung:              | Erfolgreiche Teilnahme an den Übungen und<br>Bestehen einer Abschlussprüfung  |
| Medienformen:                               | Folien / Tafel / Lehrbuch   |
| Literatur:                                  | Russell & Norvig: Artificial Intelligence – A Modern<br>Approach  |

|                                |   |
|--------------------------------|---|
| Modulbezeichnung:              | <b>Konsistenzmodelle für parallele und<br/>verteilte Systeme (IKPV)</b> |
| ggf. Modulniveau:              |   |
| ggf. Kürzel                    | IKPV  |
| ggf. Lehrveranstaltungen:      |   |
| Studiensemester:               | Ab 3. Semester Bachelor, ab 1. Semester Master                          |
| Modulverantwortliche(r):       | Studiendekan/ Studiendekanin der Informatik                             |
| Dozent(in):                    | Prof. Dr. A. Reuter   |
| Sprache:                       | Deutsch / Englisch nach Vereinbarung                                    |
| Zuordnung zum Curricu-<br>lum: |   |
| Lehrform/ SWS:                 | Vorlesung 2 SWS   |

|                                       |  |
|---------------------------------------|--|
| Arbeitsaufwand:                       | 60 h insgesamt, davon<br>30 h Präsenzstudium<br>15 h Prüfungsvorbereitung<br>15 h Selbststudium und Aufgabenbearbeitung  |
| Kreditpunkte:                         | 2 LP   |
| Voraussetzungen nach Prüfungsordnung: | Keine  |
| Empfohlene Vorkenntnisse:             | Kenntnisse zu Algorithmen und Datenstrukturen, Datenbanksysteme  |
| Lernergebnisse:                       | Die Studierenden bekommen ein Verständnis für die Probleme, die bei der Ausführung nebenläufiger Berechnungen auf gemeinsamen Ressourcen entstehen. Sie lernen die verschiedenen Ansätze zur Lösung dieser Probleme kennen und bekommen Kriterien dafür an die Hand, wie für einen gegebenen Anwendungsbereich die beste Balance zwischen Systemleistung und Konsistenz gefunden werden kann.          |
| Inhalt:                               | Klassen verteilter und paralleler Algorithmen; Gleichzeitigkeit in verteilten Systemen; Arten der Wechselwirkung zwischen parallelen Aktivitäten; die Rolle gemeinsamer Ressourcen; die Rolle replizierter Ressourcen; relevante Fehlerklassen; Möglichkeiten zur Definition von Konsistenz (syntaktisch/semantisch); Arten konsistenz erhaltender Ausführungen; Abwägung Konsistenz vs. Parallelität. |
| Studien-/ Prüfungsleistung:           | Erfolgreiche Teilnahme an einer mündlichen Prüfung   |
| Medienformen:                         | Folien / Tafel / E-Learning / Lehrbuch   |
| Literatur:                            | z.B. Distributed Systems: Concepts and Design, by G. Coulouris, J. Dollimore and T. Kindberg (Addison Wesley Press)<br>Transaction Processing-Concepts and Techniques, by Jim Gray and Andreas Reuter, Morgan Kaufmann Publishers<br>Atomic Transactions, by Nancy Lynch et al, Elsevier Publishers  |

|                           |  |
|---------------------------|--|
| Modulbezeichnung:         | <b>Logik in der Informatik (ILI)</b>           |
| ggf. Modulniveau:         |  |
| ggf. Kürzel:              | ILI  |
| ggf. Lehrveranstaltungen: |  |
| Studiensemester:          | Ab 3. Semester Bachelor, ab 1. Semester Master |

|                                       |   |
|---------------------------------------|---|
| Modulverantwortliche (r):             | Studiendekan/ Studiendekanin der Informatik   |
| Dozent(in):                           | Prof. Dr. Wolfgang Schönfeld  |
| Sprache:                              | Deutsch   |
| Zuordnung zum Curriculum:             |   |
| Lehrform/ SWS:                        | Vorlesung 2 SWS   |
| Arbeitsaufwand:                       | 90 h<br>30 h Präsenzstudium<br>10 h Prüfungsvorbereitung<br>50 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)   |
| Kreditpunkte:                         | 3 LP  |
| Voraussetzungen nach Prüfungsordnung: |   |
| Empfohlene Vorkenntnisse:             | keine   |
| Lernergebnisse:                       | Die Studierenden haben Grundkenntnisse in mathematischer Logik, wissen wo diese in der Informatik eingesetzt werden, wissen wie Kalküle aufgebaut sind, können mit diesen Fragen logisch korrekt beantworten kennen die Grundeigenschaften von Kalkülen |
| Inhalt:                               | Syntax und Semantik der Aussagenlogik<br>Syntax und Semantik der Prädikatenlogik<br>Kalküle (Sequenzenkalkül, Tableaulkalkül)<br>Beweissuchverfahren und ihre Implementierung   |
| Studien-/ Prüfungsleistung:           | Bestehen einer Abschlussprüfung   |
| Medienformen:                         | elektronische Folien, Tafel, Bücher   |
| Literatur:                            | U.Schöning, Logik für Informatiker , Spektrum Akademischer Verlag 2000<br>J. Kelly, Logik im Klartext, Pearson Studium 2003   |

|                           |   |
|---------------------------|---|
| Modulbezeichnung:         | <b>Masterarbeit (IMa)</b>                   |
| ggf. Modulniveau:         |   |
| ggf. Kürzel:              | IMa   |
| ggf. Lehrveranstaltungen: |   |
| Studiensemester:          | 4. Semester Master                          |
| Modulverantwortliche(r):  | Studiendekan/ Studiendekanin der Informatik |
| Dozent(in):               | Dozenten/ Dozentinnen der Informatik        |
| Sprache:                  | Deutsch                                     |
| Zuordnung zum             |   |

|                                       |  |
|---------------------------------------|--|
| Curriculum:                           |  |
| Lehrform/ SWS:                        | Betreutes Selbststudium 2 SWS, Kolloquium 1 SWS  |
| Arbeitsaufwand:                       | 900 h, davon<br>810 h Bearbeitung eines individuellen Themas (Forschungs- und Entwicklungsarbeiten) und schriftliche Ausarbeitung<br>90 h Ausarbeitung Vortrag und Präsentation und Mitwirkung Kolloquium  |
| Kreditpunkte:                         | 30   |
| Voraussetzungen nach Prüfungsordnung: | Keine  |
| Empfohlene Vorkenntnisse:             | Wahlpflichtvorlesungen und Module IS und IFP   |
| Lernergebnisse:                       | Einsatz der erlernten Fachkenntnisse und Methoden zum selbstständigen Lösen einer komplexen Problemstellung aus der Informatik und ihren Anwendungen<br>Fähigkeit, in großem Umfang selbstständig eine anspruchsvolle wissenschaftlichen Arbeit zu erstellen<br>Fähigkeit, eine eigene, anspruchsvolle Arbeiten in einem wissenschaftlichen Vortrag darzustellen |
| Inhalt:                               | selbstständiges wissenschaftliches Bearbeiten einer beschränkten Aufgabenstellung aus der Informatik und ihren Anwendungen<br>wissenschaftlicher Vortrag über die Problemstellung und die erarbeitete Lösung   |
| Studien-/ Prüfungsleistung:           | regelmäßige Treffen mit der/dem BetreuerIn, schriftliche Ausarbeitung und Präsentation   |
| Medienformen:                         |  |
| Literatur:                            |  |

|                           |  |
|---------------------------|--|
| Modulbezeichnung:         | <b>Management und Analyse von Datenströmen (IMADS)</b> |
| ggf. Modulniveau:         |  |
| ggf. Kürzel:              | IMADS  |
| ggf. Lehrveranstaltungen: |  |
| Studiensemester:          | Ab 3. Semester Bachelor, ab 1. Semester Master         |
| Modulverantwortliche(r):  | Studiendekan/Studiendekanin der Informatik             |
| Dozent(in):               | Prof. Dr. Michael Gertz                                |
| Sprache:                  | Deutsch/Englisch nach Vereinbarung                     |
| Zuordnung zum             |  |

|                                     |  |
|-------------------------------------|--|
| Curriculum:                         |  |
| Lehrform / SWS:                     | Vorlesung 2 SWS, Übungen 1 SWS   |
| Arbeitsaufwand:                     | 120 h; davon<br>30 h Präsenzstudium<br>15 h Prüfungsvorbereitung<br>75 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)  |
| Kreditpunkte:                       | 4 LP   |
| Voraussetzung nach Prüfungsordnung: | keine  |
| Empfohlene Vorkenntnisse:           | Algorithmen und Datenstrukturen (IAD), Datenbanken 1 (DB1)   |
| Lernergebnisse:                     | <p>Dieses Modul vertieft die Grundkenntnisse und -fähigkeiten der Datenanalyse insbesondere im Umgang mit Datenströmen.</p> <p>Die Studierenden</p> <p>kennen die Herausforderungen der Datenstromverarbeitung und sind in der Lage, gängige Methoden anzuwenden, um Datenströme zu verwalten und auszuwerten.</p> <p>wissen, welche Mining Techniken für Datenströme im Kontext von verschiedenen Domänen anwendbar sind</p> <p>kennen die unter Inhalt angegebenen Methoden und Technologien</p> |
| Inhalt:                             | <p>Methoden des Managements und der Analyse von Datenströmen.</p> <p>Datenstrommodelle</p> <p>Datenstrommanagementsysteme</p> <p>Anfragesprachen und Anfrageverarbeitung</p> <p>Synopsenbildung und Approximation</p> <p>Data Mining Techniken wie beispielsweise Clustering, Frequent Pattern Mining und Erkennung von Ausreißern</p> <p><b>Mining und Analyse von Graph-Strömen</b></p>  |
| Studien- / Prüfungsleistung:        | Erfolgreicher Abschluss eines Projekts und Bestehen einer Abschlussprüfung   |
| Medienformen:                       | Folien / Tafel / E-Learning / Lehrbuch   |
| Literatur:                          | Wissenschaftliche Veröffentlichungen. Details werden in der Vorlesung und auf der Moodle-Webseite bekannt gegeben.   |

|                                       |   |
|---------------------------------------|---|
| Modulbezeichnung:                     | <b>Mixed-Integer Programming and Combinatorial Optimization (IMIP)</b>  |
| ggf. Modulniveau:                     |   |
| ggf. Kürzel:                          | IMIP  |
| ggf. Lehrveranstaltungen:             |   |
| Studiensemester:                      | Ab 3. Semester Bachelor, ab 1. Semester Master  |
| Modulverantwortliche (r):             | Studiendekan/ Studiendekanin der Informatik   |
| Dozent(in):                           | Prof. Dr. Gerhard Reinelt   |
| Sprache:                              | Englisch  |
| Zuordnung zum Curriculum:             |   |
| Lehrform/ SWS:                        | 1-wöchiger Kompaktkurs  |
| Arbeitsaufwand:                       | 60h<br>20 h Präsenzstudium<br>10 h Prüfungsvorbereitung<br>30 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen) am Rechner   |
| Kreditpunkte:                         | 3 LP  |
| Voraussetzungen nach Prüfungsordnung: |   |
| Empfohlene Vorkenntnisse:             | Grundkenntnisse Mathematik  |
| Lernergebnisse:                       | Die Studierenden kennen die Modellierungsmöglichkeiten der gemischt-ganzzahligen Optimierung sind mit einer kommerziellen MIP-Software vertraut haben einen guten Überblick über die Anwendungsmöglichkeiten sind mit den polyedertheoretischen Grundlagen der Optimierung vertraut kennen die wichtigsten Optimierungstechniken haben die erforderlichen Kenntnisse zur Beurteilung von Möglichkeiten und Grenzen der praktischen Verwendbarkeit der erlernten Techniken |
| Inhalt:                               | Linear programming (polyhedral theory, duality, simplex algorithms, postoptimal analysis)<br>Mixed-integer modelling<br>Combinatorial optimization problems<br>Computation of optima (cutting plane, branch-and-bound. Branch-and-cut algorithms, dynamic programming)<br>Polyhedral combinatorics<br>Combinatorial polytopes   |

|                             |   |
|-----------------------------|---|
|                             | Relaxations (LP, semidefinite, Lagrange)<br>Implementation of branch-and-cut algorithms<br>Valid inequalities<br>Presolve<br>Techniques for large problems (column generation, decomposition)<br>Applications |
| Studien-/ Prüfungsleistung: | Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung  |
| Medienformen:               | Folien / Tafel / Lehrbuch   |
| Literatur:                  | z.B. Kallrath, J. and Wilson, J.M.: Business Optimisation using Mathematical Programming, Macmillan Press, 1997<br>Williams, H.P.: Model Building, Wiley, 1999  |

|                                       |   |
|---------------------------------------|---|
| Modulbezeichnung:                     | <b>Numerik von Transportprozessen in porösen Medien (INTPM)</b>   |
| ggf. Modulniveau:                     |   |
| ggf. Kürzel:                          | INTPM   |
| ggf. Lehrveranstaltungen:             |   |
| Studiensemester:                      | Ab 3. Semester Bachelor, ab 1. Semester Master  |
| Modulverantwortliche(r):              | Studiendekan/ Studiendekanin der Informatik   |
| Dozent(in):                           | Dr. Olaf Ippisch  |
| Sprache:                              | Deutsch   |
| Zuordnung zum Curriculum:             |   |
| Lehrform/ SWS:                        | Vorlesung 2 SWS, Übung 2 SWS am Rechner.  |
| Arbeitsaufwand:                       | 180 h   |
| Kreditpunkte:                         | 6 LP  |
| Voraussetzungen nach Prüfungsordnung: |   |
| Empfohlene Vorkenntnisse:             |   |
| Lernergebnisse:                       | Die Studierenden sind in der Lage<br>Strömungsprozesse in porösen Medien zu modellieren<br>Transport gelöster Stoffe zu modellieren<br>Grenzen und Voraussetzungen dieser Modelle zu verstehen<br>die wichtigsten numerischen Methoden zur Simulation dieser Modelle anzuwenden<br>die vermittelten Modelle auf konkrete Probleme der |

|                             |  |
|-----------------------------|--|
|                             | Bodensanierung anzuwenden  |
| Inhalt:                     | Grundlagen partieller Differentialgleichungen<br>Grundlagen der Strömungsmechanik<br>Grundwasserströmung und Darcy-Gesetz<br>Heterogenitäten und deren Charakterisierung, präferentielle Fließpfade<br>Transport gelöster Stoffe<br>hydrodynamische Dispersion<br>Finite Differenzen, Finite Volumen und Finite Elemente Verfahren<br>iterative Lösung linearer Gleichungssysteme. |
| Studien-/ Prüfungsleistung: | Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung (diese kann bei geringer Teilnehmerzahl durch eine mündliche Prüfung ersetzt werden).   |
| Medienformen:               | Folien/ Tafel/ E-Learning/ Lehrbuch  |
| Literatur:                  |  |

|                                       |  |
|---------------------------------------|--|
| Modulbezeichnung:                     | <b>Objekterkennung und automatisches Bildverstehen (IOAB)</b>  |
| ggf. Modulniveau:                     |  |
| ggf. Kürzel:                          | IOAB   |
| ggf. Lehrveranstaltungen:             |  |
| Studiensemester:                      | Ab 3. Semester Bachelor, ab 1. Semester Master   |
| Modulverantwortliche(r):              | Professoren des HCI  |
| Dozent(in):                           | Professoren des HCI  |
| Sprache:                              | Deutsch  |
| Zuordnung zum Curriculum:             |  |
| Lehrform/SWS:                         | Vorlesung 4 SWS , Übung 2 SWS  |
| Arbeitsaufwand:                       | 240 h  |
| Kreditpunkte:                         | 8 LP   |
| Voraussetzungen nach Prüfungsordnung: |  |
| Empfohlene Vorkenntnisse:             |  |
| Lernergebnisse:                       | Die Studierenden sind auf dem neuesten Stand der Forschung im Bereich von Mid- und High-Level Computer Vision kennen die relevantesten Ansätze zur Merkmalsextraktion sind vertraut mit den wesentlichen Methoden zur robusten Objektrepräsentation haben essentielle Algorithmen aus dem Bereich der Mustererkennung und des maschinellen |

|                             |  |
|-----------------------------|--|
|                             | <p>Lernens verstanden und können diese auf neue Probleme übertragen und anwenden<br/>sind in der Lage aktuelle Forschungsarbeiten aus dem Themenbereich der Vorlesung zu verstehen und in den Kontext der klassischen Arbeiten einzuordnen<br/>können Objekterkennungsprobleme in neuen Anwendungen analysieren und die jeweils geeignete Algorithmik dazu abrufen um darauf aufbauend neue Lösungsansätze zu entwickeln<br/>sind fähig Objekterkennungsverfahren differenziert zu bewerten und die Validität einer experimentellen Evaluation zu überprüfen</p> |
| Inhalt:                     | <p>Methoden und Modelle aus dem Bereich von Mid- und High-Level Computer Vision.<br/>Insbesondere:<br/>Objektdetektion und –klassifikation<br/>lokale und globale Merkmalsextraktion<br/>Modell-basierte Ansätze<br/>Ansichten-basierte Methoden<br/>generative/diskriminative Verfahren<br/>Registrierung<br/>Formanalyse<br/>Voting Methoden<br/>Hashing Verfahren<br/>Hierarchische Objektrepräsentationen<br/>Erkennung von menschlichen Aktionen<br/>Lernverfahren für Objektrepräsentationen</p>   |
| Studien-/ Prüfungsleistung: | Erfolgreiche Teilnahme an den Übungen und Bestehen einer Abschlussprüfung.   |
| Medienformen:               | Folien/ Tafel/ E-Learning/ Lehrbuch  |
| Literatur:                  |  |

|                           |   |
|---------------------------|---|
| Modulbezeichnung:         | <b>Objektorientiertes Programmieren im Wissenschaftlichen Rechnen (IOPWR)</b> |
| ggf. Modulniveau:         |   |
| ggf. Kürzel:              | IOPWR   |
| ggf. Lehrveranstaltungen: |   |
| Studiensemester:          | Ab 3. Semester Bachelor, ab 1. Semester Master                                |
| Modulverantwortliche (r): | Studiendekan/ Studiendekanin der Informatik                                   |
| Dozent(in):               | Prof. Dr. Peter Bastian   |
| Sprache:                  | Deutsch   |
| Zuordnung zum Curriculum: |   |

|                                       |   |
|---------------------------------------|---|
| Lehrform/SWS:                         | Vorlesung 2 SWS, Übung 2 SWS am Rechner.  |
| Arbeitsaufwand:                       | 180 h   |
| Kreditpunkte:                         | 6 LP  |
| Voraussetzungen nach Prüfungsordnung: |   |
| Empfohlene Vorkenntnisse:             | Kenntnisse einer objektorientierten Programmiersprache  |
| Lernergebnisse:                       | Die Studierenden beherrschen die Programmiersprache C++ sind in der Lage Performanz von unterschiedlichen Lösungen zu beurteilen beherrschen Template Programmieretechniken können die Standard Template Library einsetzen sind in der Lage die gelehrtten Konzepte an ausgewählten Problemen des Wissenschaftlichen Rechnens praktisch umzusetzen  |
| Inhalt:                               | Dieses Modul vertieft die in der Grundvorlesung "Einführung in die Praktische Informatik" vermittelten Kenntnisse in objektorientierter Programmierung mit spezieller Ausrichtung auf das Wissenschaftliche Rechnen:<br>Klassenkonzept<br>Dynamische Speicherverwaltung<br>Ausnahmebehandlung<br>Ressourcenallokierung und Initialisierung<br>Benutzung von const<br>Template Metaprogrammierung,<br>statischer vs. dynamischer Polymorphismus<br>Traits<br>Policies<br>Standard Template Library |
| Studien-/ Prüfungsleistung:           | Erfolgreiche Teilnahme an den Übungen (Erreichen von 50% der Punkte) und Bestehen einer Abschlussprüfung.   |
| Medienformen:                         | Folien/ Tafel/ E-Learning/ Lehrbuch   |
| Literatur:                            |   |

|                           |   |
|---------------------------|---|
| Modulbezeichnung:         | <b>Paralleles Höchstleistungsrechnen (IPHR)</b> |
| ggf. Modulniveau:         |   |
| ggf. Kürzel:              | IPHR  |
| ggf. Lehrveranstaltungen: |   |
| Studiensemester:          | Ab 3. Semester Bachelor, ab 1. Semester Master  |
| Modulverantwortliche(r):  | Studiendekan/ Studiendekanin der Informatik     |
| Dozent(in):               | Prof. Dr. Peter Bastian/ Dr. Stefan Lang        |

|                                       |  |
|---------------------------------------|--|
| Sprache:                              | Deutsch  |
| Zuordnung zum Curriculum:             |  |
| Lehrform/SWS:                         | Vorlesung 4 SWS, Übung 2 SWS am Rechner mit Diskussion der Übungsaufgaben.   |
| Arbeitsaufwand:                       | 240 h  |
| Kreditpunkte:                         | 8 LP   |
| Voraussetzungen nach Prüfungsordnung: |  |
| Empfohlene Vorkenntnisse:             | Kenntnisse einer Programmiersprache (Programmieren und Softwaretechnik) und grundlegender Algorithmen (Algorithmen und Datenstrukturen (IAD)).   |
| Lernergebnisse:                       | Die Studierenden kennen die grundlegenden Architekturen paralleler Höchstleistungsrechner<br>kennen die grundlegenden Synchronisationsmechanismen in parallelen Systemen inklusive Performanzaspekten<br>beherrschen die wichtigsten Programmierparadigmen für parallele Systeme<br>sind in der Lage grundlegende Synchronisationsaufgaben zu lösen<br>beherrschen Algorithmen der parallelen linearen Algebra<br>können die Qualität paralleler Algorithmen und deren praktische Implementierung bewerten   |
| Inhalt:                               | Rechner mit globalem Adressraum<br>Cache-Kohärenz<br>Rechner mit lokalem Adressraum und Nachrichtenaustausch<br>kritischer Abschnitt, Bedingungsynchronisation<br>Semaphore<br>Posix Threads<br>Programmierung von Grafikkarten<br>Nachrichtenaustausch<br>MPI<br>Client-Server Architekturen, entfernter Prozeduraufruf<br>Bewertung paralleler Algorithmen<br>Lastverteilung<br>Algorithmen für vollbesetzte Matrizen<br>Lösung dünnbesetzter Gleichungssysteme<br>Partikelmethode<br>Paralleles Sortieren |
| Studien-/ Prüfungsleistung:           | Erfolgreiche Teilnahme an den Übungen (Erreichen von 50%) und Bestehen einer Abschlussprüfung.   |

|               |                                     |
|---------------|-------------------------------------|
| Medienformen: | Folien/ Tafel/ E-Learning/ Lehrbuch |
| Literatur:    |                                     |

|                                       |   |
|---------------------------------------|---|
| Modulbezeichnung:                     | <b>Parallele Lösung großer Gleichungssysteme (IPLGG)</b>  |
| ggf. Modulniveau:                     |   |
| ggf. Kürzel:                          | IPLGG   |
| ggf. Lehrveranstaltungen:             |   |
| Studiensemester:                      | Ab 3. Semester Bachelor, ab 1. Semester Master  |
| Modulverantwortliche(r):              | Studiendekan/ Studiendekanin der Informatik   |
| Dozent(in):                           | Prof. Dr. Peter Bastian   |
| Sprache:                              | Deutsch   |
| Zuordnung zum Curriculum:             |   |
| Lehrform/ SWS:                        | Vorlesung 4 SWS, Übung 2 SWS am Rechner mit Diskussion der Übungsaufgaben.  |
| Arbeitsaufwand:                       | 240 h   |
| Kreditpunkte:                         | 8 LP  |
| Voraussetzungen nach Prüfungsordnung: |   |
| Empfohlene Vorkenntnisse:             | Kenntnisse einer objektorientierten Programmiersprache, MA7, Kenntnisse wie sie das Modul IPHR oder ICC1 vermittelt, Vorkenntnisse in Numerik, insbesondere Differentialgleichungen.  |
| Lernergebnisse:                       | Die Studierenden beherrschen die Diskretisierung elliptischer Differentialgleichungen mittels Finiter Elemente verstehen das abstrakte Konzept der Teilraumkorrekturverfahren können dies im Rahmen von überlappenden und nichtüberlappenden Gebietszerlegungsverfahren sowie Mehrgitterverfahren umsetzen beherrschen die Konvergenztheorie zu diesen Verfahren können ausgewählte Verfahren praktisch auf Parallelrechnern umsetzen und der Leistungsfähigkeit bewerten |
| Inhalt:                               | Grundlagen der Diskretisierung elliptischer partieller Differentialgleichungen, Teilraumkorrekturverfahren überlappende und nichtüberlappende Gebietszerlegungsverfahren mit Konvergenztheorie geometrische Mehrgitterverfahren mit   |

|                                |   |
|--------------------------------|---|
|                                | Konvergenztheorie<br>algebraische Mehrgitterverfahren   |
| Studien-/<br>Prüfungsleistung: | Erfolgreiche Teilnahme an den Übungen (Erreichen von 50% der Punkte) und Bestehen einer Abschlussprüfung. |
| Medienformen:                  | Folien/ Tafel/ E-Learning/ Lehrbuch   |
| Literatur:                     |   |

|   |  |
|---|--|
| Modulbezeichnung:                           | <b>IT-Projektmanagement (IPM)</b>  |
| ggf. Modulniveau:                           |  |
| ggf. Kürzel:                                | IPM  |
| ggf.<br>Lehrveranstaltungen:                |  |
| Studiensemester:                            | Ab 3. Semester Bachelor, ab 1. Semester Master   |
| Modulverantwortliche<br>(r):                | Studiendekan/ Studiendekanin der Informatik  |
| Dozent(in):                                 | Dr. Privat-Doz. Andrea Herrmann  |
| Sprache:                                    | Deutsch  |
| Zuordnung zum<br>Curriculum:                |  |
| Lehrform/ SWS:                              | Vorlesung+Übung 2 SWS  |
| Arbeitsaufwand:                             | 60 h insgesamt, davon<br>30 h Präsenzstudium<br>15 h Prüfungsvorbereitung<br>15 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)   |
| Kreditpunkte:                               | 2 LP   |
| Voraussetzungen<br>nach<br>Prüfungsordnung: |  |
| Empfohlene<br>Vorkenntnisse:                | Vorlesung und Übung „Einführung in Software Engineering“ (ISW) oder vergleichbare Vorkenntnisse  |
| Lernergebnisse:                             | Die Teilnehmer/innen können ein Projekt planen und überwachen, verstehen wie Projekte in Organisationen eingebettet sind und haben Grundkenntnisse in vertraglichen Themen.                                |
| Inhalt:                                     | Projektplanung, Projektorganisation<br>Kostenschätzung<br>Angebot/ Vertrag, Verhandeln<br>Vorgehensmodelle<br>Risikomanagement<br>Controlling<br>IT-Vertragsrecht<br>Änderungsmanagement<br>Zeitmanagement |

|                                |  |
|--------------------------------|--|
|                                | Projektabschluss<br>Verteilte Softwareentwicklung  |
| Studien-/<br>Prüfungsleistung: | Erfolgreiche Teilnahme an den Hausaufgaben<br>(mehr als 50% der Punkte müssen erreicht werden)<br>und Bestehen einer Abschlussprüfung. |
| Medienformen:                  | Folien & Tafel   |
| Literatur:                     | PMI (Project Management Institute): A Guide to the<br>Project Management Body of Knowledge (PM BOK<br>® Guide), 4. Ausgabe 2008        |

|  |  |
|--|--|
| Modulbezeichnung:                        | <b>Räumliche Datenbanken (IRDB)</b>  |
| ggf. Modulniveau:                        |  |
| ggf. Kürzel:                             | IRDB   |
| ggf.<br>Lehrveranstaltungen:             |  |
| Studiensemester:                         | Ab 5. Semester Bachelor, ab 1. Semester Master   |
| Modulverantwortliche(r):                 | Studiendekan/Studiendekanin der Informatik   |
| Dozent(in):                              | Prof. Dr. Michael Gertz  |
| Sprache:                                 | Deutsch/Englisch nach Vereinbarung   |
| Zuordnung zum<br>Curriculum:             |  |
| Lehrform/ SWS:                           | Vorlesung 4 SWS, Übungen 2 SWS   |
| Arbeitsaufwand:                          | 240 h; davon<br>90 h Präsenzstudium<br>15 h Prüfungsvorbereitung<br>135 h Selbststudium und Aufgabenbearbeitung<br>(eventuell in Gruppen)  |
| Kreditpunkte:                            | 8 LP   |
| Voraussetzungen nach<br>Prüfungsordnung: |  |
| Empfohlene<br>Vorkenntnisse:             | Algorithmen und Datenstrukturen (IAD),<br>Datenbanken 1 (IDB1)   |
| Lernergebnisse:                          | Die Studierenden<br>kennen die Prinzipien von und Anforderungen an<br>räumliche Daten und die Verwaltung solcher Daten<br>in verschiedenen Anwendungsbereichen (z.B. in<br>der Geographie, Kosmologie und Biologie)<br>kennen die Konzepte und Anwendungen<br>Geographischer Informationssysteme (GIS)<br>sind in der Lage, Konzepte und Modelle für<br>räumliche Daten und der Datenmodellierung<br>anzuwenden<br>sind vertraut mit der Unterstützung zur Verwaltung<br>von und Anfrage an räumliche Datenbanken (z.B.<br>PostGIS)<br>kennen grundlegende Methoden der<br>algorithmischen Geometrie und sind in der Lage, |

|                             |  |
|-----------------------------|--|
|                             | <p>entsprechende Algorithmen und Techniken anzuwenden</p> <p>kennen die wichtigsten Vertreter von Indexstrukturen zu räumlichen Daten (Gridfile, kd-Tree, Quadtree, R-Tree)</p> <p>wissen, wie in existierenden räumlichen Datenbanksystemen (z.B. PostGIS) Datenbankschemata und Anwendungen erstellt werden</p>  |
| Inhalt:                     | <p>Prinzipien und Anforderungen an die Verwaltung räumlicher Daten</p> <p>Anwendungsbereiche zur Verwaltung und Analyse räumlicher Daten</p> <p>Unterstützung zur Verwaltung räumlicher Daten in kommerziellen und Open-Source DBMS</p> <p>Konzepte und Modelle zur Repräsentation räumlicher Daten (2D, 3D); Tessellierung und Vektormodell; Gruppen von räumlichen Objekten</p> <p>Abstrakte Datentypen für räumliche Daten</p> <p>Grundlegende Techniken der algorithmischen Geometrie (z.B. Konvexe Hülle, Sweep-Line Methoden, Polygon-Partitionierung, Schnitte von Polygonen)</p> <p>Zugriffsstrukturen für räumliche Daten, insbesondere Grid-Files, kd-Tree, Quadtrees, R-Tree</p> <p>Algorithmen und Kostenmodelle für Zugriffsstrukturen zu räumlichen Daten</p> <p>Konzepte der Anfrageverarbeitung und -optimierung für räumliche Datenbanken, insb. Spatial Join</p> <p>Temporale Datenbanken und Indexstrukturen</p> <p>Moving Objects: Anwendungen, Anfragen, Indexstrukturen (B<sup>+</sup>-Tree und TPR-Tree)</p> <p>Einführung in das Mining räumlicher Daten (Clustering, Entdecken von Ausreißern)</p> <p>Überblick über Unterstützung zur Verwaltung räumlicher Daten in kommerziellen und Open-Source DBMS (PostgreSQL, MySQL, Oracle Spatial, GRASS)</p> |
| Studien-/ Prüfungsleistung: | <p>Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und erfolgreiche Teilnahme an einer schriftlichen Prüfung</p>  |
| Medienformen:               | <p>Folien/ Tafel/ E-Learning/ Lehrbuch</p>   |
| Literatur:                  | <p>z. B.: Spatial Databases – With Applications to GIS. Philippe Rigaux, Michel Scholl, Agnes Voisard. Morgan Kaufmann, 2001.</p> <p>Computational Geometry: Algorithms and Applications Mark de Berg, Otfried Cheong, Marc</p>  |

|  |   |
|--|---|
|  | van Kreveld, und Mark Overmars, Springer, Berlin, 2008.<br>Forschungsartikel aus Tagungsbänden und Journals |
|--|---|

|                                       |  |
|---------------------------------------|--|
| Modulbezeichnung:                     | <b>Seminar (IS)</b>  |
| ggf. Modulniveau:                     |  |
| ggf. Kürzel:                          | IS   |
| ggf. Lehrveranstaltungen:             |  |
| Studiensemester:                      | Ab 3. Semester Bachelor und 1.Semester Master  |
| Modulverantwortliche(r):              | Studiendekan/ Studiendekanin der Informatik  |
| Dozent(in):                           | Dozenten/ Dozentinnen der Informatik   |
| Sprache:                              | Deutsch  |
| Zuordnung zum Curriculum:             |  |
| Lehrform/ SWS:                        | Seminar 2+2 SWS  |
| Arbeitsaufwand:                       | 120 h, davon   |
| Kreditpunkte:                         | 4 LP   |
| Voraussetzungen nach Prüfungsordnung: | IPS  |
| Empfohlene Vorkenntnisse:             | Kenntnisse im Themengebiet des Seminars  |
| Lernergebnisse:                       | Kenntnis von Techniken des wissenschaftlichen Schreibens (insbesondere auch Literaturrecherche)<br>Fähigkeit, komplexe wissenschaftliche Literatur zu erschließen<br>Erweiterte Fähigkeit, komplexe wissenschaftliche Literatur in einem Vortrag zu präsentieren<br>Erweiterte Fähigkeit, zu Vorträgen zu diskutieren und Feedback zu geben<br>Fähigkeit, ein kurze wissenschaftliche Ausarbeitung zu einem komplexen Thema zu erstellen |
| Inhalt:                               | Einführung in und Einübung von Techniken des wissenschaftlichen Schreibens<br>Vertiefte Einübung der Erschließung und Präsentation wissenschaftlicher Literatur<br>Fortgeschritteneres Informatikthema   |
| Studien-/ Prüfungsleistung:           | Regelmäßige Teilnahme an den Seminarterminen und Mitwirkung in den Diskussionen<br>Ausarbeitung und Halten eines Vortrages von etwa 60 Minuten Dauer (inklusive Diskussion).<br>schriftliche Ausarbeitung von ca. 10 Seiten  |
| Medienformen:                         | Folien/ Tafel/ E-Learning/ Lehrbuch  |

|            |  |
|------------|--|
| Literatur: |  |
|------------|--|

|                                       |  |
|---------------------------------------|--|
| Modulbezeichnung:                     | <b>Simulationswerkzeuge (ISIMW)</b>  |
| ggf. Modulniveau:                     |  |
| ggf. Kürzel:                          | ISIMW  |
| ggf. Lehrveranstaltungen:             |  |
| Studiensemester:                      | Ab 3. Semester Bachelor, ab 1. Semester Master   |
| Modulverantwortliche (r):             | Prof. Dr. Peter Bastian  |
| Dozent(in):                           | Dr. Stefan Lang  |
| Sprache:                              | Deutsch  |
| Zuordnung zum Curriculum:             |  |
| Lehrform/SWS:                         | Vorlesung 2 SWS, Übung 2 SWS   |
| Arbeitsaufwand:                       | 180 h  |
| Kreditpunkte:                         | 6 LP   |
| Voraussetzungen nach Prüfungsordnung: |  |
| Empfohlene Vorkenntnisse:             | Grundlagen der Numerik   |
| Lernergebnisse:                       | <p>Die Studierenden beherrschen grundlegende Verfahren zur Repräsentation komplexer Geometrien (CAD-Modelle)</p> <p>verstehen die Methoden zur Vernetzung komplexer Geometrien</p> <p>beherrschen standardisierte Datenformate zur Speicherung großer, wissenschaftlicher Datenmengen</p> <p>haben grundlegende Verfahren zur Visualisierung von Ergebnissen von Simulationsprogrammen kennengelernt</p> <p>können diese Verfahren im Kontext realistischer Anwendungsprobleme konkret einsetzen</p> |
| Inhalt:                               | <p>Einführung in die grundlegenden Methoden, Techniken und Softwarerealisierungen bei Entwurf und Repräsentation komplexer Geometrien und deren Vernetzung. Neben der reinen Simulation eines Vorganges, d.h. der numerischen Lösung des mathematischen Problems nehmen die Vor- und Nachverarbeitungsschritte einen zunehmend größeren Stellenwert bei der Durchführung von Simulationsstudien ein. Zur Vorverarbeitung zählen z.B. die Erstellung und Repräsentation komplexer,</p>                |

|                             |   |
|-----------------------------|---|
|                             | <p>dreidimensionaler Geometrien mit Splines und Nurbs</p> <p>Zuordnung von Koeffizienten, z.B. Materialeigenschaften, und Randbedingungen</p> <p>Gittergenerierung mit Delauney und Advancing-Front-Methoden</p> <p>Speicherung größter Datenmengen, z.B. HDF5</p> <p>Datenaustausch mit Visualisierungssystemen und anderen Programmen</p> <p>Konkrete Benutzung von Simulationswerkzeugen an konkreten Beispielen, z.B. Salome, OpenCascade, Gmsh, Paraview/VTK</p> |
| Studien-/ Prüfungsleistung: | Erfolgreiche Teilnahme an den Übungen (Erreichen von 50% der Punkte aus den Übungsaufgaben).  |
| Medienformen:               | Folien/ Tafel/ E-Learning/ Lehrbuch   |
| Literatur:                  |   |

|                                       |  |
|---------------------------------------|--|
| Modulbezeichnung:                     | <b>Software Evolution (ISWEvol)</b>  |
| ggf. Modulniveau:                     |  |
| ggf. Kürzel:                          | ISWEvol  |
| ggf. Lehrveranstaltungen:             |  |
| Studiensemester:                      | Ab 4. Semester Bachelor, ab 1. Semester Master   |
| Modulverantwortliche (r):             | Prof.Dr. Barbara Paech   |
| Dozent(in):                           | Dr. Eckhart von Hahn   |
| Sprache:                              | Deutsch  |
| Zuordnung zum Curriculum:             | Angewandte Informatik sowie HörerInnen anderer Fachrichtungen  |
| Lehrform/ SWS:                        | Vorlesung 2 SWS  |
| Arbeitsaufwand:                       | <p>90h, davon</p> <ul style="list-style-type: none"> <li>• 30 h Vorlesung</li> <li>• 35 Aufgabenbearbeitung und Aufarbeitung/Selbststudium</li> <li>• 25 h Prüfungsvorbereitung (evtl. in Gruppen)</li> </ul>  |
| Kreditpunkte:                         | 3 LP   |
| Voraussetzungen nach Prüfungsordnung: |  |
| Empfohlene Vorkenntnisse:             | Kenntnisse und Fähigkeiten wie sie in Modul ISW vermittelt werden.   |
| Lernergebnisse:                       | <p>Die Studierenden können nach der Vorlesung ...</p> <ul style="list-style-type: none"> <li>• ... ein Software-Reengineering-Projekt fachlich planen und beurteilen,</li> <li>• ... bei der Ersterstellung von Software die Evolutionsfähigkeit konzeptuell sicherstellen,</li> </ul> |

|                             |   |
|-----------------------------|---|
|                             | <ul style="list-style-type: none"> <li>• ... ein Wartungskonzept für eine erstellte Software aufbauen.</li> </ul> <p>Sie kennen</p> <ul style="list-style-type: none"> <li>• ... die Unterschiede und Herausforderungen der Software-Weiterentwicklung versus der Softwareneuentwicklung – und worauf die/der InformatikerIn hierbei achten muss, sowohl aus Sicht eines Softwareherstellers als auch aus der Sicht der NutzerInnen von Software,</li> <li>• ... die klassischen Techniken der Softwareanierung,</li> <li>• ... die Typologie der Softwarewartung und das Management der Fehlerbehebung,</li> <li>• ... die Relevanz der Thematik in der Praxis der industriellen Softwareerstellung.</li> </ul>                            |
| Inhalt:                     | <p>Dieses Modul vermittelt aus konzeptioneller Sicht die Grundlagen für ein erfolgreiches Lebenszyklusmanagement von Software nach ihrer Ersterstellung. Die Vorlesungsinhalte wurden aufgrund der 15-jährigen Erfahrung des Dozenten in der Praxis zusammengestellt, auf Basis aktueller Forschung und Lehre.</p> <ul style="list-style-type: none"> <li>• Einleitung <ul style="list-style-type: none"> <li>○ Begriffsklärung, Grundlagen</li> </ul> </li> <li>• Softwareevolution <ul style="list-style-type: none"> <li>○ Softwarewartung, Softwareerhaltung</li> <li>○ Software-Reengineering</li> <li>○ Evolution und Weiterentwicklung</li> <li>○ Management der Softwareevolution</li> </ul> </li> <li>• Zusammenfassung</li> </ul> |
| Studien-/ Prüfungsleistung: | Bestehen einer mündlichen oder schriftlichen Abschlussprüfung (je nach Anzahl der Teilnehmer)   |
| Medienformen:               | Folien/ Tafel/ E-Learning   |
| Literatur:                  | <p>z.B.</p> <ul style="list-style-type: none"> <li>• Arnold, R. (Hrsg.): Software Reengineering. IEEE Computer Society Press, Los Alamitos 1993</li> <li>• Fowler, M.: Refactoring – Improving the Design of Existing Code. Addison-Wesley, Reading, Massachusetts, 1999</li> <li>• von Hahn, E.: Werterhaltung von Software. DUV, Wiesbaden 2005</li> <li>• Müller, B.: Reengineering. Eine Einführung. Teubner, Stuttgart 1997</li> <li>• Sneed, H.M.; Hasitschka, M.; Teichmann, M.-T.: Software-Produktmanagement. Wartung und Weiterentwicklung bestehender Anwendungssysteme. dpunkt, Heidelberg 2005</li> </ul>  |

- Smith, D.D.: Designing Maintainable Software. Springer, Heidelberg 1999

|                                       |  |
|---------------------------------------|--|
| Modulbezeichnung:                     | <b>Qualitätsmanagement (ISWQM)</b>   |
| ggf. Modulniveau:                     |  |
| ggf. Kürzel:                          | ISWQM  |
| ggf. Lehrveranstaltungen:             |  |
| Studiensemester:                      | Ab 4. Semester Bachelor, ab 1. Semester Master   |
| Modulverantwortliche(r):              | Studiendekan/ Studiendekanin der Informatik  |
| Dozent(in):                           | Prof. Dr. Barbara Paech  |
| Sprache:                              | Deutsch  |
| Zuordnung zum Curriculum:             |  |
| Lehrform/ SWS:                        | Vorlesung 3 SWS, Übung 3SWS  |
| Arbeitsaufwand:                       | 240 h, davon <ul style="list-style-type: none"> <li>• 90 h Präsenzstudium</li> <li>• 15 h Prüfungsvorbereitung</li> <li>• 135 h Selbststudium und Aufgabenbearbeitung im Team</li> </ul>   |
| Kreditpunkte:                         | 8 LP   |
| Voraussetzungen nach Prüfungsordnung: | Kenntnisse und Fähigkeiten wie sie in Modul ISW vermittelt werden  |
| Empfohlene Vorkenntnisse:             | Kenntnisse und Fähigkeiten wie sie in Modul ISW vermittelt werden  |
| Lernergebnisse:                       | Dieses Modul vertieft die Grundkenntnisse und –fähigkeiten des Software Engineering insbesondere im Bereich des Qualitätsmanagement.<br>Kenntnis der unter Inhalt angegebenen Methoden, Prozess und Werkzeuge<br>Fähigkeit, Qualitätssicherung, Qualitätsmanagement und Prozessverbesserung und -management einem Softwareentwicklungsprojekt durchzuführen bzw unter Anleitung angewandte Forschung dazu durchzuführen.<br>Fähigkeit Teilaufgaben im Team durchzuführen (eventuell mit „echten“ Kunden) |
| Inhalt:                               | Methoden, Prozesse und Werkzeuge für Qualitätssicherung<br>Qualitätsmanagement<br>Verbesserung von Softwareentwicklungsprozessen<br>Management von Softwareentwicklungsprozessen   |

|                                |   |
|--------------------------------|---|
| Studien-/<br>Prüfungsleistung: | Erfolgreiche Teilnahme an den Übungen (durch Erwerb einer entsprechenden Punktzahl) und erfolgreiche Teilnahme an der mündlichen oder schriftlichen Abschlussprüfung. |
| Medienformen:                  | Folien/ Tafel/ E-Learning/ Lehrbuch   |
| Literatur:                     | Wird jährlich aktualisiert  |

|  |  |
|--|--|
| Modulbezeichnung:                        | <b>Requirements Engineering (ISWRE)</b>  |
| ggf. Modulniveau:                        |  |
| ggf. Kürzel:                             | ISWRE  |
| ggf.<br>Lehrveranstaltungen:             |  |
| Studiensemester:                         | Ab 4. Semester Bachelor, ab 1. Semester Master   |
| Modulverantwortliche(r):                 | Studiendekan/ Studiendekanin der Informatik  |
| Dozent(in):                              | Prof. Dr. Barbara Paech  |
| Sprache:                                 | Deutsch  |
| Zuordnung zum<br>Curriculum:             |  |
| Lehrform/ SWS:                           | Vorlesung 3 SWS, Übung 3SWS  |
| Arbeitsaufwand:                          | 240 h, davon <ul style="list-style-type: none"> <li>• 90 h Präsenzstudium</li> <li>• 15 h Prüfungsvorbereitung</li> <li>• 135 h Selbststudium und Aufgabenbearbeitung im Team</li> </ul>   |
| Kreditpunkte:                            | 8 LP   |
| Voraussetzungen nach<br>Prüfungsordnung: | Kenntnisse und Fähigkeiten wie sie in Modul ISW vermittelt werden  |
| Empfohlene<br>Vorkenntnisse:             | Kenntnisse und Fähigkeiten wie sie in Modul ISW vermittelt werden  |
| Lernergebnisse:                          | Dieses Modul vertieft die Grundkenntnisse und –fähigkeiten des Software Engineering insbesondere im Bereich des Requirements Engineering.<br>Kenntnis der unter Inhalt angegebenen Methoden, Prozess und Werkzeuge<br>Fähigkeit, Unternehmensmodellierung, Anforderungserhebung und –verhandlung und Requirements Management in einem Softwareentwicklungsprojekt durchzuführen bzw unter Anleitung angewandte Forschung dazu durchzuführen.<br>Fähigkeit Teilaufgaben im Team durchzuführen (eventuell mit „echten“ Kunden) |
| Inhalt:                                  | Methoden, Prozesse und Werkzeuge für Unternehmensmodellierung<br>Prozessverbesserung in Unternehmen<br>Anforderungserhebung und -verhandlung   |

|                             |   |
|-----------------------------|---|
|                             | Requirements Management (Verbreitung, Prüfung und Aktualisierung von Anforderungen)   |
| Studien-/ Prüfungsleistung: | Erfolgreiche Teilnahme an den Übungen (durch Erwerb einer entsprechenden Punktzahl) und erfolgreiche Teilnahme an der mündlichen oder schriftlichen Abschlussprüfung. |
| Medienformen:               | Folien/ Tafel/ E-Learning/ Lehrbuch   |
| Literatur:                  | Wird jährlich aktualisiert  |

|                                       |  |
|---------------------------------------|--|
| Modulbezeichnung:                     | <b>Visualisierung im Bereich Cultural Heritage (IVCH)</b>  |
| ggf. Modulniveau:                     |  |
| ggf. Kürzel:                          | IVCH   |
| ggf. Lehrveranstaltungen:             |  |
| Studiensemester:                      | Ab 4. Semester Bachelor, ab 1. Semester Master   |
| Modulverantwortliche(r):              | Studiendekan/ Studiendekanin der Informatik  |
| Dozent(in):                           | Dr. Susanne Krömker  |
| Sprache:                              | Englisch oder Deutsch  |
| Zuordnung zum Curriculum:             |  |
| Lehrform/ SWS:                        | Vorlesung 2 SWS  |
| Arbeitsaufwand:                       | 60 h; davon 30 h Präsenzstudium, 30 h Prüfungsvorbereitung   |
| Kreditpunkte:                         | 2 LP   |
| Voraussetzungen nach Prüfungsordnung: |  |
| Empfohlene Vorkenntnisse:             | Einführung in die Praktische Informatik (IPR), Programmierkurs (IPK), Computergraphik 1 (ICG1)   |
| Lernergebnisse:                       | Die Studierenden sind mit unterschiedlichen Scantechniken vertraut, können Georadardaten interpretieren, sie wissen um die ethischen Grundsätze bei der Rekonstruktion, Befund und Hypothese (London Charter), kennen die Herangehensweise mit 2D und 3D Bildverarbeitung zur Erkennung von Merkmalen (Schrift). |
| Inhalt:                               | Weißlicht- und Time-of-flight-Scanner, Rekonstruktionen von Gefäßen und Gebäuden, 3D-Puzzle, Skelettierung, ethische Grundsätze  |
| Studien-/ Prüfungsleistung:           | Teilnahme an den Vorlesungen und Bestehen einer mündlichen oder schriftlichen Abschlussprüfung (je nach Teilnehmerzahl)  |
| Medienformen:                         | Folien / Lehrbuch  |
| Literatur:                            | Clive Orton: Mathematics in Archaeology. Cambridge, MA, Cambridge University Press, 1982<br>Katsushi Ikeuchi, Daisuke Miyazaki (editors): Digitally Archiving Cultural Objects. Springer, 2007<br>Christian Hörr. Algorithmen zur automatisierten  |

|  |  |
|--|--|
|  | Dokumen- tation und Klassifikation archäologischer Gefäße. Dissertation, TU Chemnitz, 2011 |
|--|--|

|                                       |   |
|---------------------------------------|---|
| Modulbezeichnung:                     | <b>Verteilte Datenbanken und Informationssysteme (IVDB)</b>   |
| ggf. Modulniveau:                     |   |
| ggf. Kürzel:                          | IVDB  |
| ggf. Lehrveranstaltungen:             |   |
| Studiensemester:                      | Ab 5. Semester Bachelor oder 1.Semester Master  |
| Modulverantwortliche(r):              | Studiendekan/Studiendekanin der Informatik  |
| Dozent(in):                           | Prof. Dr. Michael Gertz   |
| Sprache:                              | Deutsch/Englisch nach Vereinbarung  |
| Zuordnung zum Curriculum:             |   |
| Lehrform/ SWS:                        | Vorlesung 4 SWS, Übungen 2 SWS  |
| Arbeitsaufwand:                       | 240 h; davon<br>90 h Präsenzstudium<br>15 h Prüfungsvorbereitung<br>135 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)  |
| Kreditpunkte:                         | 8 ECTS  |
| Voraussetzungen nach Prüfungsordnung: | Keine   |
| Empfohlene Vorkenntnisse:             | Datenbanken 1 (IDB1), Datenbanken 2 (IDB2)  |
| Lernergebnisse:                       | Die Studierenden sind in der Lage, eine verteilte Datenbank zu entwerfen und zu realisieren<br>sind in der Lage, verschiedene Entwurfsstrategien für verteilte Datenbanken auf verschiedene Anwendungsszenarien anzuwenden<br>kennen die Techniken und Prinzipien der verteilten Anfragebearbeitung und Anfrageoptimierung<br>verstehen die Grundkonzepte und Protokolle zum Management verteilter Transaktionen, der Mehrbenutzersynchronisation und der Verwaltung von Datenreplikas<br>wissen, welche Grundkonzepte und Techniken Multi-Datenbanken zugrunde liegen<br>wissen, welche Techniken und Architekturen für die Datenverwaltung and Anfragebearbeitung im Rahmen verschiedener P2P Systeme existieren<br>können die wichtigsten Konzepte, die den mobilen Datenbanken zugrunde liegen, darstellen<br>kennen Verfahren zur verteilten Verwaltung von und Anfragebearbeitung auf Nicht-Standard Daten, |

|                             |   |
|-----------------------------|---|
|                             | wie z.B. geographische Daten oder große XML Datenmengen   |
| Inhalt:                     | <p>Grundbegriffe: verteilte Datenverarbeitung, Konzepte verteilter Datenbanken, Rechnernetzwerke</p> <p>Architekturen verteilter Datenbankmanagementsysteme (VDBMS): Transparenz, Client/Server Modelle</p> <p>Verteilter Datenbankentwurf: Entwurfsmethoden, Fragmentierung, Allokation, Integritätsbedingungen</p> <p>Anfragebearbeitung: Zielsetzungen, Relational Algebra, Ebenen der Anfragebearbeitung, Anfrage-Dekomposition und Datenlokalisierung</p> <p>Anfrageoptimierung: zentrale Optimierung, Join Algorithmen, verteilte Optimierung von Anfragen (z.B. Ingres)</p> <p>Verteilte Transaktionsverwaltung und Mehrbenutzersynchronisation: Eigenschaften und Arten von Transaktionen, Serialisierbarkeit, Concurrency Control, Locking-Strategien</p> <p>Management und Verwaltung von Daten-Replikas</p> <p>Multi-Datenbanksysteme (MDBS): Probleme heterogener MDBS, Integrationsstrategien, Architekturen.</p> <p>Grundlagen des P2P Datenmanagement</p> <p>Nicht-Standard Systeme zur verteilten Datenverwaltung: verteiltes Information Retrieval, mobile Datenbanken, verteilte Geodatenbanken, Scientific Workflows, Grid-Datenbanken</p> |
| Studien-/ Prüfungsleistung: | Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und erfolgreiche Teilnahme an einer schriftlichen Prüfung  |
| Medienformen:               | Folien/ Tafel/ E-Learning/ Lehrbuch   |
| Literatur:                  | <p>z. B.: Tamer Ozsu and Patrick Valduriez: Principles of Distributed Database Systems, 2nd edition, Prentice Hall, 1999.</p> <p>Weitere Literatur wird vom Dozenten bekannt gegeben.</p>   |

|                           |  |
|---------------------------|--|
| Modulbezeichnung:         | <b>Verteilte Systeme I (IVS1)</b>              |
| ggf. Modulniveau:         |  |
| ggf. Kürzel:              | IVS1   |
| ggf. Lehrveranstaltungen: |  |
| Studiensemester:          | Ab 3. Semester Bachelor oder 1.Semester Master |

|                                       |   |
|---------------------------------------|---|
| Modulverantwortliche(r):              | Studiendekan/ Studiendekanin der Informatik   |
| Dozent(in):                           | Prof. Dr. Artur Andrzejak und Mitarbeiter/in  |
| Sprache:                              | Deutsch   |
| Zuordnung zum Curriculum:             |   |
| Lehrform/SWS:                         | Vorlesung 2 SWS, Übung 2 SWS am Rechner mit Diskussion der Übungsaufgaben.  |
| Arbeitsaufwand:                       | 180 h   |
| Kreditpunkte:                         | 6 LP  |
| Voraussetzungen nach Prüfungsordnung: |   |
| Empfohlene Vorkenntnisse:             | Kenntnisse in Java (z.B. durch Einführung in Software Engineering (ISW)) sowie in Betriebssystemen (z.B. IBN)   |
| Lernergebnisse:                       | <p>Verständnis der unterschiedlichen parallelen Architekturen und Besonderheiten von verteilten Systemen</p> <p>Kenntnis der grundlegenden theoretischen Probleme und Algorithmen in verteilten Systemen (z.B. Datenaufteilung auf Prozessoren)</p> <p>Kenntnis über Anwendbarkeit diverser paralleler Frameworks (Pthreads, MPI, Map-Reduce) für eine gegebene Architektur</p> <p>Fähigkeit zur Erstellung von parallelen und verteilten Programmen in Java (z.B. mit Threads, RMI) oder Groovy (GPar)</p> <p>Fähigkeit der praktischen Verwendung von „modernen“ Ansätzen der parallelen/verteilten Programmierung (Map-Reduce, Actors, datenstromorientierte Ansätze)</p> <p>Vertrautheit mit Praxisfällen (Web-Dienste, parallele Verarbeitung von Daten)</p> |
| Inhalt:                               | <p>Das Modul behandelt die grundlegenden Prinzipien und Techniken der parallelen und verteilten Systeme im Kontext von Netzwerkanwendungen und der parallelen Verarbeitung von Daten. Es werden Konzepte aus den Bereichen Architekturen, Protokolle, Programmierung, Softwareframeworks und Algorithmen vorgestellt. Ein wesentlicher Teil der Vorlesung widmet sich der parallelen und verteilten Programmierung, insbesondere moderneren Ansätzen wie Map-Reduce, Actors, datenstromorientierte Programmierung.</p> <p>Ergänzende Themen umfassen Fehlertoleranz, effiziente Protokolle und Skalierbarkeit. Die Umsetzung in die Praxis erfolgt an Beispielen aus den Bereichen Web-Dienste oder Verarbeitung großer Datenmengen. Das Modul soll die</p>       |

|                             |  |
|-----------------------------|--|
|                             | Studierenden befähigen, Spezifika und Probleme der verteilten Systeme zu verstehen, verteilte Anwendungen zu programmieren und Softwareframeworks wie Hadoop effizient einzusetzen.  |
| Studien-/ Prüfungsleistung: | Erfolgreiche Teilnahme an den Übungen (Erreichen einer Mindestpunktzahl) und Bestehen einer Abschlussprüfung.  |
| Medienformen:               | Folien/ Tafel/ E-Learning/ Lehrbuch  |
| Literatur:                  | <p>Andrew S. Tanenbaum, Maarten Van Steen: Distributed Systems: Principles and Paradigms, Prentice Hall, 2006;</p> <p>George Coulouris, Jean Dollimore, Tim Kindberg: Distributed Systems: Concepts and Design (4th ed.), Addison-Wesley, 2005.</p> <p>Kurose, James F., Keith W. Ross: Computernetzwerke : der Top-Down-Ansatz, Pearson Studium, 2008.</p> <p>Grama, A., Gupta, A., Karypis, G., Kumar V.: Introduction to Parallel Computing, Addison-Wesley, 2. Auflage, 2003.</p> <p>Casanova, H.; Legrand, A.; Robert, Y.: Parallel Algorithms, Chapman &amp; Hall/CRC, 2008.</p> |

|                                       |  |
|---------------------------------------|--|
| Modulbezeichnung:                     | <b>Visualisierung in Natur- und Technikwissenschaften (IVNT)</b>   |
| ggf. Modulniveau:                     |  |
| ggf. Kürzel:                          | IVNT   |
| ggf. Lehrveranstaltungen:             |  |
| Studiensemester:                      | Ab 3. Semester Bachelor, ab 1. Semester Master   |
| Modulverantwortliche(r):              | Studiendekan/ Studiendekanin der Informatik  |
| Dozent(in):                           | Prof. Dr. Heike Leite  |
| Sprache:                              | Deutsch  |
| Zuordnung zum Curriculum:             |  |
| Lehrform/ SWS:                        | Vorlesung 2 SWS, Übung 1 SWS   |
| Arbeitsaufwand:                       | 120 h<br>45 h Präsenzstudium<br>15 h Prüfungsvorbereitung<br>60 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen) |
| Kreditpunkte:                         | 4 LP   |
| Voraussetzungen nach Prüfungsordnung: |  |

|                             |   |
|-----------------------------|---|
| Empfohlene Vorkenntnisse:   | Grundlagen der wissenschaftlichen Visualisierung  |
| Lernergebnisse:             | <p>Die Studierenden kennen die Verfahren zur Aufnahme und Simulation von wissenschaftlichen Daten in den Natur- und Technikwissenschaften.</p> <p>kennen verschiedene Formate zur Speicherung wissenschaftlicher Daten und deren interne Repräsentation innerhalb von Visualisierungssoftware.</p> <p>sind mit der Volumenvisualisierung vertraut und können die Algorithmen selbständig implementieren.</p> <p>verstehen die wesentlichen Algorithmen zur effizienten Darstellung und Analyse von Vektorfeldern.</p> <p>kennen die Verfahren zur topologischen Analyse von wissenschaftlichen Felddaten.</p> <p>kennen Algorithmen zur visuellen Analyse von multivariaten Daten und Zeitreihen.</p> |
| Inhalt:                     | <p>Datenakquise und -repräsentation in Biologie und Medizin</p> <p>Volumenvisualisierung</p> <p>Datensimulation und -repräsentation in Umwelt- und Technikwissenschaften</p> <p>Effiziente Vektorfeldvisualisierung</p> <p>Topologische Verfahren in der Visualisierung</p> <p>Visualisierung von multivariaten Daten und Zeitreihen</p>  |
| Studien-/ Prüfungsleistung: | Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen oder mündlichen Abschlussprüfung  |
| Medienformen:               | Folien / Tafel / Skript / Lehrbuch  |
| Literatur:                  | <p>C. Hanson, C. Johnson: The Visualization Handbook, Elsevier, 2005.</p> <p>R. Fernando: GPU Gems, NVidia Corp., 2004.</p> <p>LaMothe: Tricks of the 3D Game Programming Gurus — Advanced 3D Graphics and Rasterization, Sams Publications, 2003.</p> <p>Aktuelle Fachveröffentlichungen</p>   |

|                   |                                    |
|-------------------|------------------------------------|
| Modulbezeichnung: | <b>Volumenvisualisierung (IVV)</b> |
| ggf. Modulniveau: |                                    |
| ggf. Kürzel:      | IVV                                |

|  |   |
|--|---|
| ggf.<br>Lehrveranstaltungen:             |   |
| Studiensemester:                         | Ab 3. Semester Bachelor, ab 1. Semester Master  |
| Modulverantwortliche(r):                 | Studiendekan/ Studiendekanin der Informatik   |
| Dozent(in):                              | Prof. Dr. Jürgen Hesser   |
| Sprache:                                 | Englisch  |
| Zuordnung zum<br>Curriculum:             |   |
| Lehrform/ SWS:                           | Vorlesung 2 SWS, Übung 3 SWS  |
| Arbeitsaufwand:                          | 240 h<br>75 h Präsenzstudium<br>15 h Prüfungsvorbereitung<br>150 h Selbststudium und Aufgabenbearbeitung (evtl.<br>in Gruppen)  |
| Kreditpunkte:                            | 8 LP  |
| Voraussetzungen nach<br>Prüfungsordnung: |   |
| Empfohlene<br>Vorkenntnisse:             | Einführung in die Praktische Informatik,<br>Programmierkurs, Algorithmen und Datenstrukturen  |
| Lernergebnisse:                          | Die Studierenden<br>verstehen, wie experimentelle Daten zustande<br>kommen und welche prinzipiellen<br>Informationselemente diese enthalten, die für eine<br>wissenschaftliche Visualisierung notwendig sind. Sie<br>lernen die Prinzipien kennen, wie man zwischen<br>kontinuierlichen (wirkliche Verteilung) und diskreten<br>Signalen (Darstellung im Computer) wechseln kann.<br>Zudem werden sie eingeführt in Methoden der<br>Konversion von Oberflächendaten in Volumendaten<br>und umgekehrt. Schließlich werden ausgehend von<br>theoretischen physikalischen Prinzipien sie in der<br>Lage sein, verschiedene Visualisierungstechniken<br>abzuleiten und die Näherungen, die hierfür gemacht<br>werden, zu verstehen.<br>Sie sind damit in der Lage, komplexe<br>volumenorientierte Daten adäquat mathematisch zu<br>repräsentieren, zu transformieren und die<br>wesentlichen Strukturen mit adäquat darauf<br>angepassten Verfahren der Visualisierung<br>darzustellen. |
| Inhalt:                                  | Einführung in die Visualisierung wissenschaftli-<br>cher Daten aus den Natur- und Lebenswissen-<br>schaften.<br>Diskrete und kontinuierliche Repräsentation von<br>Daten sowie numerische und computergraphi-<br>sche Methoden der Interpolation.<br>Methoden der Konversion von Oberflächenreprä-<br>sentationen in Volumenrepräsentationen und<br>umgekehrt, Optimierungstechniken für effiziente   |

|                             |  |
|-----------------------------|--|
|                             | Algorithmen<br>Ableitung und Varianten von Volumenvisualisierungstechniken<br>Beschleunigungsverfahren und Parallelisierungsmethoden für Volumenvisualisierung<br>Programmieretechnik: GPU-Programmierung  |
| Studien-/ Prüfungsleistung: | Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen oder mündlichen Abschlussprüfung   |
| Medienformen:               | Folien/Lehrbuch  |
| Literatur:                  | z.B.<br>Engel et al: Real-Time Volume Graphics<br><a href="http://www.real-time-volume-graphics.org">www.real-time-volume-graphics.org</a> ,<br>Schroeder et al: VTK Textbook<br><a href="http://www.kitware.com/products/books/vtkbook.html">http://www.kitware.com/products/books/vtkbook.html</a> |

|                                       |  |
|---------------------------------------|--|
| Modulbezeichnung:                     | <b>Wissenschaftliches Arbeiten (IWA)</b>   |
| ggf. Modulniveau:                     |  |
| ggf. Kürzel:                          | IWA  |
| ggf. Lehrveranstaltungen:             |  |
| Studiensemester:                      | 1. Semester Master   |
| Modulverantwortliche(r):              | Studiendekan/Studiendekanin der Informatik   |
| Dozent(in):                           | Dozenten/Dozentinnen der Informatik  |
| Sprache:                              | Deutsch  |
| Zuordnung zum Curriculum:             |  |
| Lehrform/ SWS:                        | 2 SWS  |
| Arbeitsaufwand:                       | 60 h; davon<br>30 h Präsenzstudium<br>30 h Selbststudium und praktische Übungen<br>(eventuell in Gruppen)  |
| Kreditpunkte:                         | 2 LP   |
| Voraussetzungen nach Prüfungsordnung: |  |
| Empfohlene Vorkenntnisse:             |  |
| Lernergebnisse:                       | Die Studierenden kennen die wichtigsten Literaturquellen der Informatik wissen, welche Tools und Techniken zur Verwaltung von Literatur existieren und wie diese verwendet werden sind in der Lage, wissenschaftliche Texte (z.B. aus Tagungsbänden oder Journals) kritisch zu lesen und zu bewerten |

|                             |   |
|-----------------------------|---|
|                             | <p>kennen die fortgeschritten Techniken der Präsentation eines wissenschaftlichen Vortrags können strukturiert, verständlich, klar und schlüssig vor jedem Publikum präsentieren (Fachkompetenz) Kenntnis zentraler Erfolgsfaktoren, die für die Erstellung einer wissenschaftlichen Arbeit und die Durchführung einer Präsentation wichtig sind.</p> <p>(Methodenkompetenz) Kenntnis der Herangehensweise, einzelner Techniken und Hilfestellungen des wissenschaftlichen Arbeitens und der Erstellung von Präsentationen.</p>   |
| Inhalt:                     | <p>Wesensmerkmale einer wissenschaftlichen Arbeit, Planung und Vorbereitung der Erstellung, Hilfsmittel der Informationsgewinnung, Literaturrecherche, formale und sprachliche Anforderungen</p> <p>Aufbau und Strukturierung der Arbeit, Zitierweise, Gestaltung von Darstellungen, Literaturverzeichnis, Anhang</p> <p>Vorbereitung einer Präsentation, Analyse der Vortragssituation, Strukturierung einer Präsentation, Gestaltung des Manuskriptes, sprachliche Anforderungen: Verständlichkeit, Artikulation, Modulation; Blickkontakt, Gestik, Mimik, Körperhaltung, Visualisierung, Medieneinsatz, Interaktion mit Zuhörern</p> |
| Studien-/ Prüfungsleistung: | Regelmäßige Teilnahme, Teilnahme an praktischen Übungen und Rollenspielen   |
| Medienformen:               | Folien/ Tafel/ E-Learning/ Lehrbuch   |
| Literatur:                  |   |

***Module, die auch im Bachelor/Master Mathematik angeboten werden:***

|                           |                                  |
|---------------------------|----------------------------------|
| Modulbezeichnung:         | <b>Mathematische Logik (MB9)</b> |
| ggf. Modulniveau:         |                                  |
| ggf. Kürzel:              | MB9                              |
| ggf. Lehrveranstaltungen: |                                  |
| Studiensemester:          | Ab 4. Semester                   |
| Modulverantwortliche(r):  |                                  |
| Dozent(in):               |                                  |
| Sprache:                  |                                  |
| Zuordnung zum             |                                  |

|                                       |   |
|---------------------------------------|---|
| Curriculum:                           |   |
| Lehrform/SWS:                         | Vorlesung 4 SWS, Übung 2 SWS  |
| Arbeitsaufwand:                       | 240h  |
| Kreditpunkte:                         | 8 LP  |
| Voraussetzungen nach Prüfungsordnung: | keine   |
| Empfohlene Vorkenntnisse:             | Lineare Algebra I (MA4), Einführung in praktische Informatik (IPI)  |
| Lernergebnisse:                       | Einführung in die verschiedenen Teilgebiete der Mathematischen Logik.   |
| Inhalt:                               | Prädikatenlogik: Untersuchung der in der Mathematik üblichen logischen Schlussweisen.<br>Mengenlehre: Grundlagentheorie der Mathematik sowie Theorie der Ordinal- und Kardinalzahlen.<br>Modelltheorie: Zusammenhang zwischen axiomatischen Theorien und ihren Modellen mit Beispielen aus der Algebra.<br>Berechenbarkeitstheorie: Eigenschaften des Begriffes der berechenbaren Funktion.<br>Beweistheorie: Grenzen der Formalisierbarkeit, Unvollständigkeit und Unentscheidbarkeit. |
| Studien-/ Prüfungsleistung:           | Lösung von Übungsaufgaben mit benoteter Klausur bzw. mündlicher Prüfung. Art und Zeitpunkt einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.   |
| Medienformen:                         |   |
| Literatur:                            | Bekanntgabe in der Vorlesung  |

Entnommen aus den Modulbeschreibungen BA Mathematik vom 18.07.2011

|                                       |  |
|---------------------------------------|--|
| Modulbezeichnung:                     | <b>Wahrscheinlichkeitstheorie (MC4)</b>          |
| ggf. Modulniveau:                     |  |
| ggf. Kürzel:                          | MC4  |
| ggf. Lehrveranstaltungen:             |  |
| Studiensemester:                      | 4. Semester                                      |
| Modulverantwortliche(r):              |  |
| Dozent(in):                           |  |
| Sprache:                              |  |
| Zuordnung zum Curriculum:             |  |
| Lehrform/SWS:                         | Vorlesung 4 SWS, Übung 2 SWS                     |
| Arbeitsaufwand:                       | 240h   |
| Kreditpunkte:                         | 8 LP   |
| Voraussetzungen nach Prüfungsordnung: | keine  |
| Empfohlene                            | Analysis I und II (MA1, MA2) , Lineare Algebra I |

|                            |   |
|----------------------------|---|
| Vorkenntnisse:             | und II (MA4, MA5), Höhere Analysis (MA3), Einführung in die Wahrscheinlichkeitstheorie und Statistik (MA 8)   |
| Lernergebnisse:            | Grundlagen für alle Gebiete der Wahrscheinlichkeitstheorie und Statistik.   |
| Inhalt:                    | <p>Maß- und Integrationstheorie: <math>\sigma</math>-Algebren, Borel-<math>\sigma</math>-Algebra, messbare Abbildungen, Konstruktion von Wahrscheinlichkeitsmaßen, Produkträume. Erwartungswert als Maßintegral, Sätze von Lebesgue, Beppo Levi, Fubini und Radon-Nikodym.</p> <p>Konvergenz von Zufallsvariablen: <math>L_p</math>-Räume, Zusammenhang zwischen fast sicherer, stochastischer und <math>L_p</math>-Konvergenz, Starkes Gesetz der großen Zahlen, Konvergenz in Verteilung, charakteristische Funktionen, zentraler Grenzwertsatz.</p> <p>Bedingte Verteilungen: Bedingte Erwartungen, Markov-Kerne, Martingale in diskreter Zeit.</p> <p>Stochastische Prozesse: Brownsche Bewegung, Poisson-Prozess, Empirischer Prozess.</p> |
| Studien-/Prüfungsleistung: | Lösung von Übungsaufgaben mit benoteter 2-stündiger Klausur. Art und Zeitpunkt einer Wiederholungsprüfung wird vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.   |
| Medienformen:              |   |
| Literatur:                 | <ul style="list-style-type: none"> <li>- Bauer, H.: Wahrscheinlichkeitstheorie, de Gruyter</li> <li>- Billingsley, P.: Probability and Measure, Wiley.</li> <li>- Dudley, R.N.: Real Analysis and Probability</li> <li>- Durrett, S.: Probability: Theory and Examples, Duxbury Press</li> <li>- Jacod, J. and Protter, P.: Probability Essentials, Springer</li> <li>- Shiryaev, A.: Probability, Springer.</li> </ul>   |

Entnommen aus den Modulbeschreibungen BA Mathematik vom 18.07.2011

|                           |                      |
|---------------------------|----------------------|
| Modulbezeichnung:         | <b>Numerik (MD1)</b> |
| ggf. Modulniveau:         |                      |
| ggf. Kürzel:              | MD1                  |
| ggf. Lehrveranstaltungen: |                      |
| Studiensemester:          | 3. Semester          |
| Modulverantwortliche(r):  |                      |
| Dozent(in):               |                      |
| Sprache:                  |                      |

|                                       |  |
|---------------------------------------|--|
| Zuordnung zum Curriculum:             |  |
| Lehrform/SWS:                         | Vorlesung 4 SWS, Übung 2 SWS   |
| Arbeitsaufwand:                       | 240 h  |
| Kreditpunkte:                         | 8 LP   |
| Voraussetzungen nach Prüfungsordnung: | keine  |
| Empfohlene Vorkenntnisse:             | Analysis 1 (MA1), Lineare Algebra 1 (MA4), Einführung in die Numerik (MA7)   |
| Lernergebnisse:                       | Kenntnisse der numerischen Lösung von Anfangswert- und Randwertaufgaben gewöhnlicher Differentialgleichungen und einfacher partieller Differentialgleichungen.   |
| Inhalt:                               | Theorie von Anfangs- und Randwertaufgaben<br>Einschrittmethoden: Konsistenz, Stabilität, Konvergenz.<br>Numerische Stabilität und steife Anfangswertaufgaben<br>Andere Verfahrensklassen: Lineare Mehrschrittmethoden, Extrapolationsmethoden, Galerkin-Methoden (optional).<br>Lösung von Differentiellalgebraischen Aufgaben<br>Lösung von Randwertaufgaben: Schießverfahren, Differenzen- und Galerkin-Verfahren (optional).<br>Differenzenverfahren für elliptische partielle Differentialgleichungen, Laplace-Gleichung, 5-Punkte-Approximation.<br>Iterative Lösungsverfahren für diskretisierte Probleme. |
| Studien-/ Prüfungsleistung:           | Lösung von Übungsaufgaben mit benoteten 2-stündigen Klausuren, Wiederholungsmöglichkeit mit der Vorlesung im Folgejahr.  |
| Medienformen:                         |  |
| Literatur:                            | Bekanntgabe in der Vorlesung (Vorlesungsskriptum)  |

Entnommen aus den Modulbeschreibungen BA Mathematik vom 18.07.2011

|                           |                        |
|---------------------------|------------------------|
| Modulbezeichnung:         | <b>Statistik (MD2)</b> |
| ggf. Modulniveau:         |                        |
| ggf. Kürzel:              | MD2                    |
| ggf. Lehrveranstaltungen: |                        |
| Studiensemester:          | 4. Semester            |
| Modulverantwortliche(r):  |                        |
| Dozent(in):               |                        |
| Sprache:                  |                        |
| Zuordnung zum             |                        |

|                                       |  |
|---------------------------------------|--|
| Curriculum:                           |  |
| Lehrform/SWS:                         | Vorlesung 4 SWS, Übung 2 SWS   |
| Arbeitsaufwand:                       | 240h   |
| Kreditpunkte:                         | 8 LP   |
| Voraussetzungen nach Prüfungsordnung: | keine  |
| Empfohlene Vorkenntnisse:             | Analysis I (MA1), Lineare Algebra I (MA4), Einführung in die Wahrscheinlichkeitstheorie u. Statistik (MA8), Wahrscheinlichkeitstheorie (MC4)   |
| Lernergebnisse:                       | Prinzipien der mathematischen Statistik  |
| Inhalt:                               | Entscheidungstheorie: Dualität von Tests und Konfidenzbereichen, Neyman-Pearson-Theorie, allgemeine Entscheidungsverfahren, Risikofunktionen, Bayes- und Minimaxoptimalität<br>Asymptotische Statistik: Verteilungsapproximation, Fisher-Information, relative asymptotische Effizienz von Tests und Schätzern, Likelihood-basierte Verfahren, nichtparametrische Verfahren. |
| Studien-/ Prüfungsleistung:           | Lösung von Übungsaufgaben mit benoteter 2-stündiger Klausur. Art und Zeitpunkt einer Wiederholungsprüfung wird vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben   |
| Medienformen:                         |  |
| Literatur:                            | <ul style="list-style-type: none"> <li>- Bickel, P. J. and Doksum, K. A.: Mathematical Statistics, Prentice Hall</li> <li>- Lehmann, E. L.: Testing Statistical Hypothesis, Springer Verlag</li> <li>- Van der Vaart, A. W.: Asymptotic Statistics, Cambridge University Press</li> </ul>  |

Entnommen aus den Modulbeschreibungen BA Mathematik vom 18.07.2011

|                                       |                                  |
|---------------------------------------|----------------------------------|
| Modulbezeichnung:                     | <b>Lineare Optimierung (MD3)</b> |
| ggf. Modulniveau:                     |                                  |
| ggf. Kürzel:                          | MD3                              |
| ggf. Lehrveranstaltungen:             |                                  |
| Studiensemester:                      |                                  |
| Modulverantwortliche(r):              |                                  |
| Dozent(in):                           |                                  |
| Sprache:                              |                                  |
| Zuordnung zum Curriculum:             |                                  |
| Lehrform/SWS:                         |                                  |
| Arbeitsaufwand:                       | 240h                             |
| Kreditpunkte:                         | 8 LP                             |
| Voraussetzungen nach Prüfungsordnung: | keine                            |

|                             |  |
|-----------------------------|--|
| Empfohlene Vorkenntnisse:   |  |
| Lernergebnisse:             |  |
| Inhalt:                     |  |
| Studien-/ Prüfungsleistung: |  |
| Medienformen:               |  |
| Literatur:                  |  |

Entnommen aus den Modulbeschreibungen BA Mathematik vom 18.07.2011

|                                       |                                       |
|---------------------------------------|---------------------------------------|
| Modulbezeichnung:                     | <b>Nichtlineare Optimierung (MD4)</b> |
| ggf. Modulniveau:                     |                                       |
| ggf. Kürzel:                          | MD4                                   |
| ggf. Lehrveranstaltungen:             |                                       |
| Studiensemester:                      |                                       |
| Modulverantwortliche(r):              |                                       |
| Dozent(in):                           |                                       |
| Sprache:                              |                                       |
| Zuordnung zum Curriculum:             |                                       |
| Lehrform/SWS:                         |                                       |
| Arbeitsaufwand:                       | 240h                                  |
| Kreditpunkte:                         | 8 LP                                  |
| Voraussetzungen nach Prüfungsordnung: | keine                                 |
| Empfohlene Vorkenntnisse:             |                                       |
| Lernergebnisse:                       |                                       |
| Inhalt:                               |                                       |
| Studien-/ Prüfungsleistung:           |                                       |
| Medienformen:                         |                                       |
| Literatur:                            |                                       |

Entnommen aus den Modulbeschreibungen BA Mathematik vom 18.07.2011

|                           |   |
|---------------------------|---|
| Modulbezeichnung:         | <b>Wissenschaftliches Rechnen (MD5)</b> |
| ggf. Modulniveau:         |   |
| ggf. Kürzel:              | MD5                                     |
| ggf. Lehrveranstaltungen: |   |
| Studiensemester:          |   |
| Modulverantwortliche(r):  |   |
| Dozent(in):               |   |
| Sprache:                  |   |
| Zuordnung zum             |   |

|                                       |   |
|---------------------------------------|---|
| Curriculum:                           |   |
| Lehrform/SWS:                         | Vorlesung 4 SWS, Übung 2 SWS  |
| Arbeitsaufwand:                       | 240h  |
| Kreditpunkte:                         | 8 LP  |
| Voraussetzungen nach Prüfungsordnung: | keine   |
| Empfohlene Vorkenntnisse:             | Mathematische Grundvorlesungen (MA1-MA8)  |
| Lernergebnisse:                       | Grundlegende Methoden der Modellbildung und Simulation. Schwerpunktmäßig werden die Prozesse betrachtet, die sich mit Hilfe partieller Differentialgleichungen beschreiben lassen.  |
| Inhalt:                               | Hauptthemen sind<br>Modellbildung: Modellierungssystematik, Diskrete Systeme, kontinuierliche Prozesse, Standardansätze zur Modellierung, Erhaltungsgleichung.<br>Simulationsmethoden: Grundlegende Diskretisierungsverfahren, elementare Lösertechniken.<br>Anwendungsbeispiele: Hier kommen einfache Anwendungen aus Biologie, Medizin, Physik u. a. zur Diskussion |
| Studien-/ Prüfungsleistung:           | Lösung von Übungsaufgaben mit benoteter Klausur bzw. mündlicher Prüfung. Art und Zeitpunkt einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.   |
| Medienformen:                         |   |
| Literatur:                            | Es wird ein Skriptum angeboten.   |

Entnommen aus den Modulbeschreibungen BA Mathematik vom 18.07.2011

|                                       |                                 |
|---------------------------------------|---------------------------------|
| Modulbezeichnung:                     | <b>Computeralgebra I (MG19)</b> |
| ggf. Modulniveau:                     |                                 |
| ggf. Kürzel:                          | MG19                            |
| ggf. Lehrveranstaltungen:             |                                 |
| Studiensemester:                      |                                 |
| Modulverantwortliche(r):              |                                 |
| Dozent(in):                           |                                 |
| Sprache:                              |                                 |
| Zuordnung zum Curriculum:             |                                 |
| Lehrform/ SWS:                        | Vorlesung 4 SWS, Übung 2 SWS    |
| Arbeitsaufwand:                       | 240 h                           |
| Kreditpunkte:                         | 8 LP                            |
| Voraussetzungen nach Prüfungsordnung: | keine                           |

|                             |  |
|-----------------------------|--|
| Empfohlene Vorkenntnisse:   | Algebra I (MB1)  |
| Lernergebnisse:             | Grundkenntnisse in Computeralgebra.  |
| Inhalt:                     | <p>Die Vorlesung Computeralgebra befasst sich mit der Theorie und der Komplexität grundlegender mathematischer Algorithmen und deren Implementierungen in Computeralgebrasystemen. Hauptthemen sind:</p> <p><i>I. Schnelle Arithmetik:</i> Komplexität der elementaren Grundoperationen, diskrete Fouriertransformation, schnelle Multiplikation und schneller Euklidischer Algorithmus, Subresultanten und Polynomrestfolgen, modulare Algorithmen, Rechnen mit algebraischen Zahlen, schnelle Matrizenmultiplikation</p> <p><i>II. Primzerlegung und Primzahltests:</i> Primzahltest von Solovay- Strassen und Miller-Rabin, der AKS-Primzahlentest, RSA-Schema, elementare Primzahlzerlegungsverfahren, quadratisches Sieb, Irreduzibilitätstest für Polynome, Berlekamp-Algorithmen, Zassenhaus-Algorithmus, Gitter-Basis-Reduktion, Faktorisierung multivariater Polynome</p> <p><i>III. Gröbnerbasen-Algorithmen:</i> Gröbnerbasen und reduzierte Gröbnerbasen, Buchberger-Algorithmus, Eliminationstheorie, Algorithmen für elementare Idealoperationen, Berechnung der Dimension eines Ideals.</p> |
| Studien-/ Prüfungsleistung: | <p>Lösung von Übungsaufgaben mit benoteter Klausur- bzw. mündlicher Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekanntgegeben.</p> <p>.</p>   |
| Medienformen:               |  |
| Literatur:                  | <ul style="list-style-type: none"> <li>- J. von zur Gathen, J. Gerhard: Modern Computer Algebra</li> <li>- O. Geddes, S. R. Czapor, G. Labahn: Algorithms for Computer Algebra</li> <li>- D. Cox, J. Little, D. O'Shea: Ideals, Varieties and Algorithms</li> <li>- B. H. Matzat: Computeralgebra (Skriptum, in Vorbereitung)</li> </ul>   |

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 18.07.2011

|                                       |   |
|---------------------------------------|---|
| Modulbezeichnung:                     | <b>Computeralgebra II (MG20)</b>  |
| ggf. Modulniveau:                     |   |
| ggf. Kürzel:                          | MG20  |
| ggf. Lehrveranstaltungen:             |   |
| Studiensemester:                      |   |
| Modulverantwortliche(r):              |   |
| Dozent(in):                           |   |
| Sprache:                              |   |
| Zuordnung zum Curriculum:             |   |
| Lehrform/SWS:                         | Vorlesung 4 SWS, Übung 2 SWS  |
| Arbeitsaufwand:                       | 240h  |
| Kreditpunkte:                         | 8 LP  |
| Voraussetzungen nach Prüfungsordnung: | keine   |
| Empfohlene Vorkenntnisse:             | Algebra I und II (MB1, MB2), Computeralgebra I (MG19)   |
| Lernergebnisse:                       | Vertiefte Kenntnisse in Computeralgebra   |
| Inhalt:                               | Die Vorlesung Computeralgebra II behandelt eines oder mehrere Gebiete aus dem folgenden Themenkatalog:<br><br>I. Algorithmische Zahlentheorie<br>II. Algorithmische kommutative Algebra<br>III. Algorithmische Gruppentheorie<br>IV. Algorithmische Invariantentheorie<br>V. Algorithmische Arithmetische Geometrie |
| Studien-/ Prüfungsleistung:           | Lösung von Übungsaufgaben mit benoteter Klausur- bzw. mündlicher Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.   |
| Medienformen:                         |   |
| Literatur:                            | Wird in der Vorlesung bekannt gegeben   |

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 18.07.2011

|                           |   |
|---------------------------|---|
| Modulbezeichnung:         | <b>Numerische Lineare Algebra (MH5)</b> |
| ggf. Modulniveau:         |   |
| ggf. Kürzel:              | MH5                                     |
| ggf. Lehrveranstaltungen: |   |
| Studiensemester:          |   |
| Modulverantwortliche(r):  |   |
| Dozent(in):               |   |

|                                       |   |
|---------------------------------------|---|
| Sprache:                              |   |
| Zuordnung zum Curriculum:             |   |
| Lehrform/SWS:                         | Vorlesung 4 SWS, Übung 2 SWS  |
| Arbeitsaufwand:                       | 240h  |
| Kreditpunkte:                         | 8 LP  |
| Voraussetzungen nach Prüfungsordnung: | keine   |
| Empfohlene Vorkenntnisse:             | Einführung in die Numerik (MA7)   |
| Lernergebnisse:                       | Kenntnis der gebräuchlichen Methoden zur numerischen Lösung von Aufgaben der Linearen Algebra   |
| Inhalt:                               | Lineare Gleichungssysteme und Eigenwertaufgaben<br>Iterative Verfahren, Fixpunktiterationen<br>Krylowraum-Methoden<br>Iterative Verfahren für Eigenwertaufgaben<br>Singularwertzerlegung<br>Anwendungen auf Systemmatrizen bei der Diskretisierung partieller Differentialgleichungen |
| Studien-/ Prüfungsleistung:           | Lösung von Übungsaufgaben mit benoteter 2-stündiger Klausur (und einer Freischussmöglichkeit), Wiederholungsmöglichkeit mit der Vorlesung in den Folgejahren.   |
| Medienformen:                         |   |
| Literatur:                            | Bekanntgabe in der Vorlesung (Vorlesungsskriptum)   |

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 18.07.2011

|                                       |   |
|---------------------------------------|---|
| Modulbezeichnung:                     | <b>Numerik gewöhnlicher Differentialgleichungen (MH6)</b> |
| ggf. Modulniveau:                     |   |
| ggf. Kürzel:                          | MH6   |
| ggf. Lehrveranstaltungen:             |   |
| Studiensemester:                      |   |
| Modulverantwortliche(r):              |   |
| Dozent(in):                           |   |
| Sprache:                              |   |
| Zuordnung zum Curriculum:             |   |
| Lehrform/SWS:                         | Vorlesung 4 SWS, Übung 2 SWS                              |
| Arbeitsaufwand:                       | 240h  |
| Kreditpunkte:                         | 8 LP  |
| Voraussetzungen nach Prüfungsordnung: | keine   |

|                             |   |
|-----------------------------|---|
| Empfohlene Vorkenntnisse:   | Grundkenntnisse in numerischer Mathematik, der linearen Algebra und Analysis sowie eine Programmiersprache, Vorlesung Gewöhnliche Differentialgleichungen (MC1)   |
| Lernergebnisse:             | Methodenkenntnis in der Numerik gewöhnlicher Differentialgleichungen  |
| Inhalt:                     | Das Modul behandelt numerische Verfahren zur Lösung von Anfangswertproblemen und Randwertproblemen bei gewöhnlichen Differentialgleichungen und differentiell-algebraischen Gleichungen. Es werden im Rahmen des Moduls Diskretisierungsverfahren (Einschritt-, Mehrschritt-, Extrapolationsverfahren), Konvergenzuntersuchungen (Konsistenz, Stabilität) sowie die praktische Realisierung in mathematischer Software (Fehlerschätzung, Ordnungs- und Schrittweitensteuerung) und ihre Anwendung vermittelt. Speziell eingegangen wird im Rahmen von Randwertproblemen auf Mehrzielverfahren und die Lösung der entstehenden großen impliziten Gleichungssysteme mit speziellen Newton-Typ Verfahren. Das Modul legt die Grundlage für die numerische Behandlung allgemeinerer Klassen von Problemstellungen (Differentialgleichungsbeschränkte Optimierungsprobleme, Optimalsteuerungsprobleme, Probleme der Parameterschätzung sowie Randwertprobleme bei partiellen Differentialgleichungen). |
| Studien-/ Prüfungsleistung: | Erfolgreiche Teilnahme an den Übungen (Erreichen einer Mindestpunktzahl) und Bestehen einer Abschlussprüfung.   |
| Medienformen:               |   |
| Literatur:                  |   |

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 18.07.2011

|                           |   |
|---------------------------|---|
| Modulbezeichnung:         | <b>Numerik partieller Differentialgleichungen (MH7)</b> |
| ggf. Modulniveau:         |   |
| ggf. Kürzel:              |   |
| ggf. Lehrveranstaltungen: |   |
| Studiensemester:          |   |
| Modulverantwortliche(r):  |   |
| Dozent(in):               |   |
| Sprache:                  |   |
| Zuordnung zum             |   |

|                                       |   |
|---------------------------------------|---|
| Curriculum:                           |   |
| Lehrform/SWS:                         | Vorlesung 4 SWS, Übung 2 SWS  |
| Arbeitsaufwand:                       | 240h  |
| Kreditpunkte:                         | 8 LP  |
| Voraussetzungen nach Prüfungsordnung: | keine   |
| Empfohlene Vorkenntnisse:             | Analysis I (MA1), Lineare Algebra I (MA4), Einführung in die Numerik (MA7), Partielle Differentialgleichungen I (MC2)   |
| Lernergebnisse:                       | Kenntnis der gebräuchlichen Methoden zur numerischen Lösung von Anfangswert- und Anfangs-Randwert-Aufgaben der prototypischen partiellen Differentialgleichungen: Laplace-, Wärmeleitungs- und Wellengleichung.   |
| Inhalt:                               | I. Theorie von partiellen Differentialgleichungen, Typeneinteilung und Lösungseigenschaften<br>II. Differenzenverfahren für elliptische Randwertaufgaben<br>III. Finite-Elemente-Verfahren für elliptische Randwertaufgaben: Diskretisierung, a posteriori Fehlerschätzung und Gitteradaption<br>IV. Lösungsverfahren für große lineare Gleichungssysteme: Fixpunktiterationen, Krylowraum- und Mehrgittermethoden<br>V. Verfahren für parabolische Anfangs-Randwertaufgaben (Wärmeleitungsgleichung)<br>VI. Verfahren für hyperbolische Probleme (Wellengleichung) |
| Studien-/Prüfungsleistung:            | Lösung von Übungsaufgaben mit benoteter 2-stündiger Klausur (und einer Freischussmöglichkeit), Wiederholungsmöglichkeit mit der Vorlesung in den Folgejahren.   |
| Medienformen:                         |   |
| Literatur:                            | Bekanntgabe in der Vorlesung (Vorlesungsskriptum)   |

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 18.07.2011

|                           |   |
|---------------------------|---|
| Modulbezeichnung:         | <b>Numerische Optimierung bei Differentialgleichungen (MH8)</b> |
| ggf. Modulniveau:         |   |
| ggf. Kürzel:              | MH8   |
| ggf. Lehrveranstaltungen: |   |
| Studiensemester:          |   |
| Modulverantwortliche(r):  |   |
| Dozent(in):               |   |
| Sprache:                  |   |

|                                       |  |
|---------------------------------------|--|
| Zuordnung zum Curriculum:             |  |
| Lehrform/ SWS:                        | Vorlesung 4 SWS, Übung 2 SWS   |
| Arbeitsaufwand:                       | 240h   |
| Kreditpunkte:                         | 8 LP   |
| Voraussetzungen nach Prüfungsordnung: | keine  |
| Empfohlene Vorkenntnisse:             | Kenntnisse, wie sie in den Vorlesungen Einführung in die Numerische Mathematik und Numerische Mathematik I vermittelt werden sowie Grundkenntnisse der linearen Algebra und Analysis sowie einer Programmiersprache, Vorlesung Gewöhnliche Differentialgleichungen (MC1) |
| Lernergebnisse:                       | Parameterschätzung sowie optimale nichtlineare Versuchsplanung bei Differentialgleichungen.  |
| Inhalt:                               | Das Modul behandelt Grundlagen und numerische Methoden der optimalen Steuerung.  |
| Studien-/ Prüfungsleistung:           | Erfolgreiche Teilnahme an den Übungen (Erreichen einer Mindestpunktzahl) und Bestehen einer Abschlussprüfung.  |
| Medienformen:                         |  |
| Literatur:                            |  |

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 18.07.2011

|                                       |  |
|---------------------------------------|--|
| Modulbezeichnung:                     | <b>Statistik II (MH12)</b>   |
| ggf. Modulniveau:                     |  |
| ggf. Kürzel:                          | MH12   |
| ggf. Lehrveranstaltungen:             |  |
| Studiensemester:                      |  |
| Modulverantwortliche(r):              |  |
| Dozent(in):                           |  |
| Sprache:                              |  |
| Zuordnung zum Curriculum:             |  |
| Lehrform/SWS:                         | Vorlesung 4 SWS, Übung 2 SWS   |
| Arbeitsaufwand:                       | 240h   |
| Kreditpunkte:                         | 8 LP   |
| Voraussetzungen nach Prüfungsordnung: | Analysis I (MA1), Lineare Algebra I (MA4)                                      |
| Empfohlene Vorkenntnisse:             | Wahrscheinlichkeitstheorie I (MC5), Statistik I (MD3)                          |
| Lernergebnisse:                       | Vertiefte Behandlung einer Auswahl statistischer Methoden                      |
| Inhalt:                               | Mögliche Themen sind:<br><i>I. Multivariate Statistik: Wishart-Verteilung,</i> |

|                             |   |
|-----------------------------|---|
|                             | <p>multipler Korrelationskoeffizient, Hotellings T<sup>2</sup> - Verteilung, Hauptkomponentenanalyse, kanonische Korrelationen, grafische Modelle</p> <p><i>II. Zeitreihenanalyse:</i> Lineare Filter, ARMA-Modelle, Prädiktion, State-Space Modelle, Spektraldarstellung, Periodogramm, Whittle-Likelihood, nichtlineare Zeitreihenmodelle</p> <p><i>III. Nichtparametrik:</i> Dichteschätzung und nichtparametrische Regression, Kernschätzer, lokal polynomiale Schätzer, Orthogonalreihenschätzer, Adaptivität, Risikoabschätzung, nichtparametrische Tests</p> |
| Studien-/ Prüfungsleistung: | Lösung von Übungsaufgaben mit benoteter 2-stündiger Klausur. Art und Zeitrahmen einer Wiederholungsprüfung wird vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.  |
| Medienformen:               |   |
| Literatur:                  | <ul style="list-style-type: none"> <li>- Anderson, T. W.: An Introduction to Multivariate Statistical Analysis, John Wiley</li> <li>- Jørgensen, Bent: The Theory of Linear Models, Chapman &amp; Hall, New York, 1993.</li> <li>- Brockwell, P. J. and Davis R. A.: Time Series: Theory and Methods, Springer-Verlag</li> <li>- Wasserman, L.: All of Nonparametric Statistics, Springer-Verlag</li> </ul>   |

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 18.07.2011

|                                       |   |
|---------------------------------------|---|
| Modulbezeichnung:                     | <b>Wahrscheinlichkeitstheorie II (MH13)</b> |
| ggf. Modulniveau:                     |   |
| ggf. Kürzel:                          | MH13  |
| ggf. Lehrveranstaltungen:             |   |
| Studiensemester:                      |   |
| Modulverantwortliche(r):              |   |
| Dozent(in):                           |   |
| Sprache:                              |   |
| Zuordnung zum Curriculum:             |   |
| Lehrform/SWS:                         | Vorlesung 4 SWS, Übung 2 SWS                |
| Arbeitsaufwand:                       | 240h  |
| Kreditpunkte:                         | 8 LP  |
| Voraussetzungen nach Prüfungsordnung: | keine                                       |
| Empfohlene                            | Analysis I (MA1), Lineare Algebra I (MA4),  |

|                             |   |
|-----------------------------|---|
| Vorkenntnisse:              | Wahrscheinlichkeitstheorie I (MC4)  |
| Lernergebnisse:             | Ausgewählte Themen zu Stochastischen Prozessen und zur Stochastischen Analyse.  |
| Inhalt:                     | <p><i>I. Theorie Stochastischer Prozesse:</i> Endlich-dimensionale Verteilungen, Existenzsatz von Kolmogorov, stetige Pfade, Konstruktion und Eigenschaften der Brownschen Bewegung, Gaußprozesse</p> <p><i>II. Ergodentheorie:</i> Stationäre und ergodische Prozesse, Ergodensätze</p> <p><i>III. Invarianzprinzipien:</i> Straffheit, schwache Konvergenz im Raum der stetigen Funktionen, Invarianzprinzip von Donsker, Theorie der Empirischen Prozesse</p> <p><i>IV. Stochastisches Integral:</i> Martingale in stetiger Zeit, Itô-Integral, Itô-Formel</p> |
| Studien-/ Prüfungsleistung: | Lösung von Übungsaufgaben mit benoteter Klausur bzw. mündlicher Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung wird vom Dozent festgelegt und zu Beginn der Vorlesung bekannt gegeben.  |
| Medienformen:               |   |
| Literatur:                  | <ul style="list-style-type: none"> <li>- Durrett, S.: Probability: Theory and Examples, Duxbury Press</li> <li>- Karlin, S. and Taylor, H.: A First/Second Course in Stochastic Processes, Academic Press</li> <li>- Karatzas, I. and Shreve, S.: Brownian Motion and Stochastic Calculus, Springer</li> </ul>  |

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 18.07.2011

|                           |   |
|---------------------------|---|
| Modulbezeichnung:         | <b>Berechenbarkeit und Komplexität I (MH14)</b> |
| ggf. Modulniveau:         |   |
| ggf. Kürzel:              | MH14  |
| ggf. Lehrveranstaltungen: |   |
| Studiensemester:          |   |
| Modulverantwortliche(r):  |   |
| Dozent(in):               |   |
| Sprache:                  |   |
| Zuordnung zum Curriculum: |   |
| Lehrform/ SWS:            | Vorlesung 4 SWS, Übungen 2 SWS                  |

|                                       |  |
|---------------------------------------|--|
| Arbeitsaufwand:                       | 240h   |
| Kreditpunkte:                         | 8 LP   |
| Voraussetzungen nach Prüfungsordnung: | keine  |
| Empfohlene Vorkenntnisse:             | Grundkenntnisse aus der Theoretischen Informatik sind hilfreich  |
| Lernergebnisse:                       | Grundkenntnisse über Berechenbarkeit und Komplexität   |
| Inhalt:                               | Die Berechenbarkeitstheorie liefert den formalen Rahmen, die Lösbarkeit algorithmischer Probleme zu untersuchen, die Komplexitätstheorie stellt Methoden und Konzepte zur Analyse des erforderlichen Aufwands algorithmischer Problemlösungen zur Verfügung. Ziel des Moduls ist es die Studierenden mit den zentralen Konzepten und Methoden der Berechenbarkeits- und der Komplexitätstheorie vertraut zu machen. In der Berechenbarkeitstheorie stehen Methoden zum Nachweis der Unentscheidbarkeit im Mittelpunkt, in der Komplexitätstheorie liegt der Schwerpunkt auf dem Vergleich und der strukturellen Analyse der polynomiell beschränkten Komplexitätsklassen. Insbesondere werden das P-NP-Problem und die NP-Vollständigkeit behandelt. |
| Studien-/ Prüfungsleistung:           | Lösung von Übungsaufgaben mit benoteter Klausur bzw. mündlicher Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.   |
| Medienformen:                         |  |
| Literatur:                            |  |

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 18.07.2011

|                           |  |
|---------------------------|--|
| Modulbezeichnung:         | <b>Berechenbarkeit und Komplexität II (MH15)</b> |
| ggf. Modulniveau:         |  |
| ggf. Kürzel:              | MH15   |
| ggf. Lehrveranstaltungen: |  |
| Studiensemester:          |  |
| Modulverantwortliche(r):  |  |
| Dozent(in):               |  |
| Sprache:                  |  |
| Zuordnung zum Curriculum: |  |
| Lehrform/ SWS:            | Vorlesung 3 SWS, Übungen 1 SWS                   |

|                                       |   |
|---------------------------------------|---|
| Arbeitsaufwand:                       | 180 h   |
| Kreditpunkte:                         | 6 LP  |
| Voraussetzungen nach Prüfungsordnung: | keine   |
| Empfohlene Vorkenntnisse:             | Berechenbarkeit und Komplexität I (MH14)  |
| Lernergebnisse:                       | Vertiefte Kenntnisse über Berechenbarkeit und Komplexität   |
| Inhalt:                               | In diesem Modul werden ausgewählte fortgeschrittene Themen aus dem Bereich der Berechenbarkeits- und Komplexitätstheorie behandelt.   |
| Studien-/ Prüfungsleistung:           | Lösung von Übungsaufgaben mit benoteter Klausur bzw. mündlicher Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben |
| Medienformen:                         |   |
| Literatur:                            |   |

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 18.07.2011

|                                       |  |
|---------------------------------------|--|
| Modulbezeichnung:                     | <b>Algorithmische Optimierung I (MH16)</b>   |
| ggf. Modulniveau:                     |  |
| ggf. Kürzel:                          | MH16   |
| ggf. Lehrveranstaltungen:             |  |
| Studiensemester:                      |  |
| Modulverantwortliche(r):              |  |
| Dozent(in):                           |  |
| Sprache:                              |  |
| Zuordnung zum Curriculum:             |  |
| Lehrform/SWS:                         | Vorlesung 4 SWS, Übung 2 SWS   |
| Arbeitsaufwand:                       | 240 h  |
| Kreditpunkte:                         | 8 LP   |
| Voraussetzungen nach Prüfungsordnung: | keine  |
| Empfohlene Vorkenntnisse:             | Mathematische Grundvorlesungen (MA1- MA8)  |
| Lernergebnisse:                       | Grundkenntnisse über algorithmische Optimierung  |
| Inhalt:                               | Das Modul behandelt moderne Verfahren der unbeschränkten und beschränkten Optimierung. Die Studierenden werden in die Lage versetzt, moderne Verfahren des Gebietes anzuwenden, zu beurteilen und zu entwickeln. |

|                                |  |
|--------------------------------|--|
| Studien-/<br>Prüfungsleistung: | Lösung von Übungsaufgaben mit benoteter Klausur bzw. mündlicher Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben. |
| Medienformen:                  |  |
| Literatur:                     | Bekanntgabe in der Vorlesung   |

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 18.07.2011

|  |  |
|--|--|
| Modulbezeichnung:                        | <b>Algorithmische Optimierung II (MH17)</b>  |
| ggf. Modulniveau:                        |  |
| ggf. Kürzel:                             | MH17   |
| ggf.<br>Lehrveranstaltungen:             |  |
| Studiensemester:                         |  |
| Modulverantwortliche(r):                 |  |
| Dozent(in):                              |  |
| Sprache:                                 |  |
| Zuordnung zum<br>Curriculum:             |  |
| Lehrform/ SWS:                           | Vorlesung 4 SWS, Übung 2 SWS   |
| Arbeitsaufwand:                          | 240h   |
| Kreditpunkte:                            | 8 LP   |
| Voraussetzungen nach<br>Prüfungsordnung: | keine  |
| Empfohlene<br>Vorkenntnisse:             | Algorithmische Optimierung I (MH16)  |
| Lernergebnisse:                          | Grundlagen der linearen und ganzzahligen Optimierung.  |
| Inhalt:                                  | I. Dualitätstheorie<br>II. Simplexalgorithmus und Varianten<br>III. Innere-Punkte-Verfahren<br>IV. Schnittebenen-Verfahren   |
| Studien-/<br>Prüfungsleistung:           | Lösung von Übungsaufgaben mit benoteter Klausur bzw. mündlicher Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben. |
| Medienformen:                            |  |
| Literatur:                               | Bekanntgabe in der Vorlesung   |

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 18.07.2011

|                   |                               |
|-------------------|-------------------------------|
| Modulbezeichnung: | <b>Mustererkennung (MH18)</b> |
|-------------------|-------------------------------|

|                                       |   |
|---------------------------------------|---|
| ggf. Modulniveau:                     |   |
| ggf. Kürzel:                          | MH18  |
| ggf. Lehrveranstaltungen:             |   |
| Studiensemester:                      |   |
| Modulverantwortliche(r):              | Studiendekan Mathematik   |
| Dozent(in):                           |   |
| Sprache:                              |   |
| Zuordnung zum Curriculum:             |   |
| Lehrform/SWS:                         | Vorlesung 4 SWS, Übung 2 SWS  |
| Arbeitsaufwand:                       | 240h  |
| Kreditpunkte:                         | 8 LP  |
| Voraussetzungen nach Prüfungsordnung: | keine   |
| Empfohlene Vorkenntnisse:             | Mathematische Grundvorlesungen (MA1- MA8), Wahrscheinlichkeitstheorie I (MC4), Numerische Lineare Algebra (MH5), Algorithmische Optimierung I (MH16)  |
| Lernergebnisse:                       | Selbständiges computergestütztes Lösen von Problemen der Statistischen Mustererkennung.   |
| Inhalt:                               | Euklidische Einbettungen, Multidimensionale Skalierung, Bayes Klassifikator, Fehlerschranken und -abschätzungen, Entscheidungsbäume, Kombination und Performanzsteigerung einfacher Klassifikatoren, Klassifikation mit Kernfunktionen, Gaußsche Prozesse und Klassifikation, Klassifikation mit Mischungsverteilungen, nichtparametrische Klassifikation, Merkmalsauswahl und -extraktion, Ballungsanalyse mit Prototypen, Ähnlichkeitsgraphen und unüberwachtes Lernen. |
| Studien-/ Prüfungsleistung:           | Bearbeiten von Übungsblättern und Übungen am Computer, Semesterbegleitende Prüfung, Art und Zeitrahmen werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben   |
| Medienformen:                         |   |
| Literatur:                            | Bekanntgabe in der Vorlesung und auf der entsprechenden WWW-Seite   |

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 18.07.2011

|                           |                                |
|---------------------------|--------------------------------|
| Modulbezeichnung:         | <b>Bildverarbeitung (MH19)</b> |
| ggf. Modulniveau:         |                                |
| ggf. Kürzel:              | MH19                           |
| ggf. Lehrveranstaltungen: |                                |

|                                       |  |
|---------------------------------------|--|
| Studiensemester:                      |  |
| Modulverantwortliche(r):              | Studiendekan Mathematik  |
| Dozent(in):                           |  |
| Sprache:                              |  |
| Zuordnung zum Curriculum:             |  |
| Lehrform/SWS:                         | Vorlesung 4 SWS, Übung 2 SWS   |
| Arbeitsaufwand:                       | 240h   |
| Kreditpunkte:                         | 8 LP   |
| Voraussetzungen nach Prüfungsordnung: | keine  |
| Empfohlene Vorkenntnisse:             | Mathematische Grundvorlesungen (MA1- MA8), Fouriertransformation, Variationsrechnung, Numerische Lineare Algebra (MH5), Partielle Differentialgleichungen II (MH3), Numerik Partieller Differentialgleichungen (MH7), Statistische Mustererkennung (MH18)  |
| Lernergebnisse:                       | Selbständiges computergestütztes Lösen von Problemen der Digitalen Bildverarbeitung.   |
| Inhalt:                               | Repräsentation der Bildfunktion (Fouriertransformation, Abtasttheorem, Wavelettransformation), Variationsmethoden zur Bildrestauration, Bildmerkmale, Textur, Optischer Fluss und Korrespondenzanalyse, Bildsegmentierung, Epipolargeometrie und Stereorekonstruktion, Objekt- und Ereigniserkennung |
| Studien-/ Prüfungsleistung:           | Bearbeiten von Übungsblättern und Übungen am Computer, Semesterbegleitende Prüfung, Art und Zeitrahmen werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben  |
| Medienformen:                         |  |
| Literatur:                            | Bekanntgabe in der Vorlesung und auf der entsprechenden WWW-Seite  |

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 18.07.2011

**Module, die im Master Technische Informatik angeboten werden:**

Alle fachlichen Module (nicht Fachübergreifende Kompetenzen) aus dem Master Technische Informatik können – entsprechend der inhaltlichen Voraussetzungen – auch im Master Angewandte Informatik belegt werden. Das Angebot findet sich im jeweils gültigen Modulhandbuch des Masters Technische Informatik.

[http://www.ziti.uniheidelberg.de/ziti/index.php?option=com\\_content&view=article&id=135&Itemid=77&lang=de](http://www.ziti.uniheidelberg.de/ziti/index.php?option=com_content&view=article&id=135&Itemid=77&lang=de)

**Module, die auch im Bachelor/ Master Physik angeboten werden**

|                                      |  |
|--------------------------------------|--|
| Modulbezeichnung:                    | <b>Physics of Imaging (MWInf5)</b>   |
| ggf. Modulniveau                     |  |
| ggf. Kürzel                          | MWInf5   |
| ggf. Untertitel                      |  |
| ggf. Lehrveranstaltungen:            |  |
| Studiensemester:                     |  |
| Modulverantwortliche (r):            |  |
| Dozent(in):                          |  |
| Sprache:                             |  |
| Zuordnung zum Curriculum             |  |
| Lehrform/ SWS:                       | Lecture on „Physics of Imaging“ (4hours/week)  |
| Arbeitsaufwand:                      | 120 h  |
| Kreditpunkte:                        | 4 LP   |
| Voraussetzungen nach Prüfungsordnung |  |
| Empfohlene Voraussetzungen:          | UKInf2, PEP1-PEP4  |
| Lernergebnisse:                      | Basics of the Physics of Imaging; common principles and techniques of imaging for atomic to astronomical scales.   |
| Inhalt:                              | <ul style="list-style-type: none"> <li>-Projective geometry, optics, wave optics, Fourier optics and lens aberrations</li> <li>-Radiometry of imaging</li> <li>-Methods of imaging: scanning electron microscopy, X-ray, EDX, FLIM,FRET , fluorescence imaging, near-field imaging</li> <li>-CCD and CMOS technology</li> <li>-Holography, ultrasound imaging, CT- computer tomography, magnetic resonance imaging</li> <li>-Satellite imaging, synthetic aperture radar, radio astronomy</li> </ul> |
| Studien-/ Prüfungsleistungen:        | Defined by lecturer before beginning of course   |
| Medienformen:                        |  |
| Literatur:                           | Announced by lecturer  |

Entnommen aus dem Modulhandbuch MScPhysik, Version 2011.1

|                   |                                  |
|-------------------|----------------------------------|
| Modulbezeichnung: | <b>Image Processing (MWInf6)</b> |
| ggf. Modulniveau  |                                  |
| ggf. Kürzel       | MWInf6                           |
| ggf. Untertitel   |                                  |

|  |  |
|--|--|
| ggf.<br>Lehrveranstaltungen:               |  |
| Studiensemester:                           |  |
| Modulverantwortliche<br>(r):               |  |
| Dozent(in):                                |  |
| Sprache:                                   |  |
| Zuordnung zum<br>Curriculum                |  |
| Lehrform/ SWS:                             | Lecture on „Image Processing“ (4hours/week)<br>Exercises (2hours/week)   |
| Arbeitsaufwand:                            | 210 h  |
| Kreditpunkte:                              | 7 LP   |
| Voraussetzungen<br>nach<br>Prüfungsordnung |  |
| Empfohlene<br>Voraussetzungen:             | UKInf1   |
| Lernergebnisse:                            | Learn how to analyze signals from time series, images, and any kind of multidimensional signals and to apply it to problems in natural sciences, life sciences and technology.   |
| Inhalt:                                    | <ul style="list-style-type: none"> <li>-Continuous and discrete signals, sampling theorem, signal representation</li> <li>-Fourier transform</li> <li>-Random variables and fields, probability density functions, error propagation</li> <li>-Homogeneous and inhomogeneous point operations</li> <li>-Neighbourhood operations, linear and nonlinear filters, linear system theory</li> <li>-Geometric transformations and interpolation</li> <li>-Multi-grid signal presentation and processing</li> <li>-Averaging, edge and line detection, local structure analysis, local phase and wave numbers</li> <li>-Motion analysis in image sequences</li> <li>-Segmentation</li> <li>-Regression, globally optimal signal analysis, variation approaches, steerable and nonlinear filtering, inverse filtering</li> <li>-Morphology and shape analysis, moments, Fourier descriptors</li> <li>-Bayesian image restoration</li> <li>--Object detection and recognition</li> </ul> |
| Studien-/<br>Prüfungsleistungen:           | Defined by lecturer before beginning of course   |
| Medienformen:                              |  |
| Literatur:                                 | B. Jähne, Digital Image Processing, 6th edition, Springer  |

|                                      |  |
|--------------------------------------|--|
| Modulbezeichnung:                    | <b>Pattern Recognition (MWInf7)</b>  |
| ggf. Modulniveau                     |  |
| ggf. Kürzel                          | MWInf7   |
| ggf. Untertitel                      |  |
| ggf. Lehrveranstaltungen:            |  |
| Studiensemester:                     |  |
| Modulverantwortliche (r):            |  |
| Dozent(in):                          |  |
| Sprache:                             |  |
| Zuordnung zum Curriculum             |  |
| Lehrform/ SWS:                       | Lecture on „Pattern Recognition “ (4hours/week)<br>Exercises (2hours/week)   |
| Arbeitsaufwand:                      | 210h   |
| Kreditpunkte:                        | 7 LP   |
| Voraussetzungen nach Prüfungsordnung |  |
| Empfohlene Voraussetzungen:          | UKInf1; Knowledge about Linear Algebra, Probability, Statistics  |
| Angestrebte Lernergebnisse:          | Given a huge bunch of data, find out what’s in it; build automated diagnostic systems or expert systems that automatically learn to make reliable predictions from a training set of examples. Lectures and exercises will be interwoven and allow you to build such systems by yourself; real-life examples will be drawn from for the application areas named below.   |
| Inhalt:                              | -Curse of dimensionality<br>-Variable selection and dimension reduction for high-dimensional data<br>-Unsupervised learning: Cluster analysis<br>-Supervised learning: Regression<br>-Supervised learning: Classification by means of neural networks, support vector machines, etc.<br>-Graphical models<br>-Applications: Data mining, industrial quality control, process monitoring, astrophysics, medicine, life sciences |
| Studien-/ Prüfungsleistungen:        | Defined by lecturer before beginning of course   |

|               |  |
|---------------|--|
| Medienformen: |  |
| Literatur:    | Pattern Classification (2nd ed.) by Richard O. Duda, Peter E. Hart. and David G. Stork. Wiley, 2000. |

Entnommen aus dem Modulhandbuch MScPhysik, Version 2011.1

|                                       |   |
|---------------------------------------|---|
| Modulbezeichnung:                     | <b>Introduction to Image Processing on the GPU (IGPU)</b>   |
| ggf. Modulniveau:                     |   |
| ggf. Kürzel:                          | IGPU  |
| ggf. Lehrveranstaltungen:             |   |
| Studiensemester:                      | From 5th semester bachelor, or 1st semester master  |
| Modulverantwortliche(r):              |   |
| Dozent(in):                           | Dr. Bastian Goldluecke  |
| Sprache:                              | English course material and exercise sheets<br>Course language English, switching to German only if there is no demand for English  |
| Zuordnung zum Curriculum:             |   |
| Lehrform/ SWS:                        | Four day block course, equivalent of 2 SWS  |
| Arbeitsaufwand:                       | 60 h<br>30 h Lectures and lab exercises<br>30 h Preparation and home exercises  |
| Kreditpunkte:                         | 2 LP  |
| Voraussetzungen nach Prüfungsordnung: |   |
| Empfohlene Vorkenntnisse:             | Solid knowledge of C programming, in particular no fear of pointers and direct memory access<br>Some basics of image processing are helpful, but not strictly necessary   |
| Lernergebnisse:                       | For certain computation tasks which are colloquially called „embarrassingly parallel“, and which occur quite frequently for example in image analysis, an implementation on a GPU (i.e. graphics card) can be orders of magnitude faster than a similar CPU implementation.<br>In this short course, the students learn:<br>- what kind of code can and can not be parallelized on a graphics processing unit |

|                            |   |
|----------------------------|---|
|                            | <p>(GPU)</p> <ul style="list-style-type: none"> <li>- the structure of a GPU, the different types of memory and how and when to use them</li> <li>- how to program the GPU using nVidia CUDA</li> <li>- how to perform various image analysis tasks using the GPU</li> <li>- ways to optimize the speed of GPU code</li> </ul>  |
| Inhalt:                    | <p>The four days will be broken up into theoretical and practical segments. In the (short) theoretical lectures, I will introduce the concepts necessary for GPU programming and basic image analysis, which you will immediately try out afterwards in lab sessions. This way, the course will be highly practical and interactive. Students will work in small groups, each of which will have the task to write a small program solving an image analysis problem of their choice. Note that while image analysis tasks are a focus of the course because they give immediately visible results, the acquired techniques can be of course be employed in other fields as well, e.g. for solving large linear algebra problems or PDEs.</p> |
| Studien-/Prüfungsleistung: | Active participation in the sessions and completion of the assigned program.  |
| Medienformen:              | Slides / blackboard / online manuals / example code   |
| Literatur:                 | e.g. nVidia CUDA Programming Guide, available in the "CUDA zone" on <a href="http://www.nvidia.com">www.nvidia.com</a>  |

|                           |  |
|---------------------------|--|
| Modulbezeichnung:         | <b>Fast Parallel Implementations of Image Labeling Problems (IMLP)</b>               |
| ggf. Modulniveau:         |  |
| ggf. Kürzel:              | IMLP   |
| ggf. Lehrveranstaltungen: |  |
| Studiensemester:          | From 5th semester bachelor, or 1st semester master                                   |
| Modulverantwortliche(r):  |  |
| Dozent(in):               | Dr. Bastian Goldluecke   |
| Sprache:                  | English course material and exercise sheets<br>Course language English, switching to |

|                                       |   |
|---------------------------------------|---|
|                                       | German only if there is no demand for English   |
| Zuordnung zum Curriculum:             |   |
| Lehrform/ SWS:                        | 2 SWS Lecture + 1 SWS Exercise  |
| Arbeitsaufwand:                       | 120 h<br>45 h Lectures and lab exercises<br>60 h Revision and home exercises<br>15 h Exam preparation   |
| Kreditpunkte:                         | 4 LP  |
| Voraussetzungen nach Prüfungsordnung: |   |
| Empfohlene Vorkenntnisse:             | Either „Introduction to Image Processing on the GPU“ offered at the beginning of the semester, or an introduction to <i>Variational Image Analysis</i> from e.g. last semester.   |
| Lernergebnisse:                       | <i>Image labeling problems</i> are a fundamental class of problems appearing in image analysis, which dominate tasks in low-level computer vision like depth and motion estimation. Recently, many algorithms have been developed to solve this kind of problems in a variational framework, which allows for fast parallel implementations on the GPU. In this short course, the students learn: <ul style="list-style-type: none"> <li>- Theoretical background for solving labeling problems in a variational framework</li> <li>- Efficient algorithms to solve the related class of optimization problems with parallel algorithms, which can be implemented on the GPU</li> <li>- How to implement these algorithms using nVidia CUDA</li> <li>- Techniques and tricks to make the implementations efficient</li> </ul> |
| Inhalt:                               | The lecture is aimed at students who either participated in my CUDA course at the beginning of the semester and want to learn more about image analysis and the theoretical background, or participated in a theoretical course on variational image analysis and want to learn more about state-of-the-art labeling algorithms and the practical side of their implementation. If you are new to both topics, you might still take part in the course, but must be prepared to take (potentially a lot of) additional time learning the prerequisites.   |

|                                |  |
|--------------------------------|--|
| Studien-/<br>Prüfungsleistung: | Active participation in lecture and exercises,<br>oral exam.   |
| Medienformen:                  | Slides / blackboard / online manuals /<br>example code   |
| Literatur:                     | e.g. Chambolle et al. 2010 "An Introduction to<br>Total Variation for Image Analysis" for<br>theoretical background, and nVidia CUDA<br>Programming Guide, available in the "CUDA<br>zone" on <a href="http://www.nvidia.com">www.nvidia.com</a> , for the practical<br>one. |

**Module, die auch im Bachelor/ Master Biologie angeboten  
werden**

|  |   |
|--|---|
| Modulbezeichnung                           | <b>Model-based segmentation of biomedical<br/>images<br/>(Modellbasierte Segmentierung<br/>Biomedizinischer Bilder, IMSBI)</b>  |
| Modulniveau                                |   |
| Kürzel                                     | IMSBI   |
| Lehrveranstaltungen                        |   |
| Studiensemester                            |   |
| Modulverantwortlicher                      | Studiendekan/Studiendekanin der Informatik  |
| Dozent                                     | Dr. Stefan Wörz   |
| Sprache                                    | Deutsch oder Englisch   |
| Zuordnung zum<br>Curriculum                |   |
| Lehrform / SWS                             | Vorlesung 2 SWS   |
| Arbeitsaufwand                             | 120h insgesamt, davon<br>30h Präsenzstudium<br>30h Prüfungsvorbereitung<br>60h Selbststudium und Aufgabenbearbeitung  |
| Kreditpunkte                               | 4 LP  |
| Voraussetzungen<br>nach<br>Prüfungsordnung |   |
| Empfohlene<br>Vorkenntnisse                | Grundkenntnisse Signal- oder Bildverarbeitung (z.B.<br>Vorlesung „Bildverarbeitung“ oder „Bioinformatik“)   |
| Lernergebnisse                             | Ziel der Vorlesung ist es, einen Überblick über 2D<br>und 3D deformierbare Modelle und deren<br>mathematische Grundlagen für die modellbasierte<br>Segmentierung biomedizinischer Bilder zu geben |
| Inhalt                                     | In der Vorlesung werden drei wesentliche Klassen<br>von deformierbaren Modellen (aktive Konturen,   |

|                            |   |
|----------------------------|---|
|                            | <p>statistische Formmodelle, analytische parametrische Modelle) sowie starre Template Modelle behandelt. Die verschiedenen Segmentierungsverfahren werden an aktuellen Beispielen aus der 2D und 3D biomedizinischen Bildverarbeitung veranschaulicht. Geplante Themen sind unter anderem:</p> <ul style="list-style-type: none"> <li>- Introduction and overview of image analysis, image segmentation, and biomedical images</li> <li>- Template models, linear transformations, Hough transform</li> <li>- Active contour models: snakes, level sets, and geodesic active contours</li> <li>- Statistical shape models: active shape models, active appearance models, and extensions</li> <li>- Shape correspondence estimation: landmark-based, intensity-based, and hybrid</li> <li>- Analytic parametric models: contour-based and intensity-based</li> <li>- Performance evaluation and applications</li> </ul> |
| Studien-/ Prüfungsleistung | Bestehen einer Abschlussprüfung   |
| Medienformen               | Folien und Tafel  |
| Literatur                  | Bekanntgabe in der Vorlesung  |

### ***Module im Anwendungsgebiet***

|                           |   |
|---------------------------|---|
| Modulbezeichnung:         | <b>Anwendungsgebiet (IAG)</b>   |
| ggf. Modulniveau:         |   |
| ggf. Kürzel:              | IAG   |
| ggf. Lehrveranstaltungen: |   |
| Studiensemester:          | Ab 1.Semester Master  |
| Modulverantwortliche(r):  | Studiendekan/ Studiendekanin der Informatik   |
| Dozent(in):               | Dozenten/ Dozentinnen der Informatik  |
| Sprache:                  | Deutsch   |
| Zuordnung zum Curriculum: |   |
| Lehrform/ SWS:            | Vorlesung, Übung und/oder Praktikum   |
| Arbeitsaufwand:           | 540 h, Aufteilung in Präsenz- / Übungs- und Praktikumszeit in Absprache mit den Dozentinnen bzw. Dozenten   |
| Kreditpunkte:             | 18 LP, davon 8 LP mindestens durch einen nicht-informatischen Modul auf Masterniveau (Ausnahmegenehmigung möglich), die restlichen 10 LP können durch ein Projekt oder Module auf |

|                                       |  |
|---------------------------------------|--|
|                                       | Bachelorniveau erbracht werden   |
| Voraussetzungen nach Prüfungsordnung: | keine  |
| Empfohlene Vorkenntnisse:             | Gleiches Anwendungsgebiet im Bachelor  |
| Lernergebnisse:                       | Vertiefte Kenntnisse und Fähigkeiten in einem Anwendungsgebiet   |
| Inhalt:                               | <p>Wahl eines Anwendungsgebietes nach den Regeln der Prüfungsordnung</p> <p>Festlegung von und Teilnahme an Modulen aus dem Anwendungsgebiet (die LP entsprechen dabei den Vorgaben aus dem Anwendungsgebiet). Dabei ist sicherzustellen, dass keine Module aus dem Anwendungsgebiet gewählt werden, die schon im Bachelor-Studium eingebracht wurden.</p> <p>(optional) Definition und Durchführung eines interdisziplinären Projektes, d.h.</p> <p>Festlegung eines Dozenten / einer Dozentin aus dem Anwendungsgebiet und der Informatik</p> <p>Gemeinsame Festlegung des Projektziels durch die Dozentinnen bzw. Dozenten und den/die Studierende. Das Projektziel umfasst eine informatische Leistung im Anwendungsgebiet.</p> <p>Festlegung des Arbeitsaufwandes und damit der LP für das Projekt</p> <p>Durchführung des Projekts entsprechend des Projektziels unter Anleitung der Dozentinnen bzw. Dozenten</p> <p>Dokumentation des Ergebnisses</p> <p>Erstellung eines Projektbericht</p> <p>Präsentation des Ergebnisses</p> |
| Studien-/ Prüfungsleistung:           | Prüfungsleistungen in dem Anwendungsgebiet und (optional) Prüfungsleistungen für das interdisziplinäre Projekt analog zum Modul IFP, gewichtet nach dem jeweiligen Anteil der LP   |
| Medienformen:                         | Folien/ Tafel/ E-Learning/ Lehrbuch  |
| Literatur:                            |  |