

Rahmenstudienplan

für den Studiengang Informatik an der Universität Hamburg

vom 7. 6. 2000

1 Vorbemerkungen

Der Rahmenstudienplan dient zur inhaltlichen Ausfüllung des reformierten Studienkonzeptes im Studiengang Informatik an der Universität Hamburg vor dem Hintergrund der ab Wintersemester 1998/99 gültigen Studien- und Prüfungsordnung.

Er dient den Studierenden zur Planung ihres Studiums, den Lehrenden zur Koordination und inhaltlichen Gestaltung des Lehrangebots. Er enthält Regelungen zu Lehrveranstaltungen und Prüfungen im Grund- und Hauptstudium und beschreibt die Wahlmöglichkeiten zu den einzelnen Studienprofilen.

Der Rahmenstudienplan wird vom Studienreformausschuß des Fachbereichs in Zusammenarbeit mit dem Prüfungsausschuß erstellt bzw. weiterentwickelt und vom Fachbereichsrat erlassen bzw. revidiert.

Die vorliegende, vom Fachbereichsrat am 7.6.2000 verabschiedete Fassung enthält gegenüber der ersten Version aus 1998 Änderungen und Ergänzungen. Geändert wurden die Inhalte des F-Zyklus im Grundstudium, leicht überarbeitet die Festlegungen zu Proseminaren. Ergänzt wurden folgende bisher fehlende Teile für das Hauptstudium:

- der Katalog der Grundlagenveranstaltungen,
- die angebotenen Studienprofile sowie ihre Schwerpunkte,
- die Festlegungen zu Seminaren, Projekten und Projektseminaren.

2 Grundstudium

Das Grundstudium besteht aus prüfungsrelevanten Pflichtveranstaltungen in der Informatik (62 SWS) und Wahlpflichtveranstaltungen im Ergänzungsfach (12SWS), für die zum Teil unbenotete Leistungsnachweise (z.B. Übungsscheine) erforderlich sind.

Dazu kommen Wahlpflichtveranstaltungen in der Informatik (12 SWS) mit unbenoteten Leistungsnachweisen oder Teilnahmebestätigungen sowie Wahlveranstaltungen nach Belieben.

2.1 Übersicht über das Grundstudium Informatik

Mathematische Grundlagen der Informatik	Formale Grundlagen der Informatik	Technische Informatik	Praktische Informatik	Informatik, Mensch und Gesellschaft	Ergänzungsfach	SWS
M1 Diskrete Mathematik <i>Klausur</i> 4V 2Ü	F1 Logik <i>Übungsschein (F1 oder F2)</i> 2V 1Ü	T1 Digitale Systeme 1 2V 1Ü	P1 Deklarative und applikative Programmierung <i>Klausur</i> 2V 2Ü	IMG1 Informatik - eine Standortbestimmung 2V	ER1 Wahl- und Orientierungsphase	18
M2 Analysis <i>Übungsschein (M2 oder M3)</i> 4V 2Ü	F2 Automaten und Kalküle <i>Klausur (F1 und F2)</i> 2V 1Ü	T2 Digitale Systeme 2 <i>Klausur (T1 und T2)</i> 2V 1Ü	P2 Zustands- und objektorientierte Programmierung <i>Übungsschein</i> 2V 2Ü		ER2 je nach Fach 4SWS	20
M3 Lineare Algebra <i>mdl. Prüfung (M2 und M3)</i> 2V 2 Ü	F3 Berechenbarkeit und Komplexität 2V	T3 Rechnerorganisation und Betriebssysteme <i>Übungsschein (T3 oder T4)</i> 2V 1Ü	P3 Algorithmen und Datenstrukturen <i>Übungsschein mdl. Prüfung (P2 und P3)</i> 2V 2Ü		ER3 je nach Fach 4SWS	17
M4 Stochastik <i>Übungsschein</i> 4V 2Ü	F4 Parallelität und Nebenläufigkeit <i>mdl. Prüfung (F3 und F4)</i> 2V	T4 Parallelverarbeitung und Kommunikationsnetze <i>mdl. Prüfung (T3 und T4)</i> 2V 1Ü	IMG3 Grundlagen natürlicher Informationsverarbeitung 2V	IMG4 Informatiksysteme in der Gesellschaft <i>mdl. Prüfung (IMG 1, 3 oder 4)</i> 2V	ER4 je nach Fach <i>Prüfung je nach Fach</i> 4SWS	19
M 14V + 8Ü = 22	F 8V + 2Ü = 10	T 8V + 4Ü = 12	P 6V + 6Ü = 12	IMG 6V = 6	ER 12	74

Wahlpflicht:	Praktikum in Praktischer Informatik	4PR	<i>Praktikumsschein</i>	4
	Praktikum in Technischer Informatik	4PR	<i>Praktikumsschein</i>	4
	Proseminar aus IMG	2PS	<i>Teilnahmebestätigung</i>	2
	anderes Proseminar	2PS	<i>Teilnahmebestätigung</i>	2

Summe Pflicht und Wahlpflicht: 86

Wahl: Einführung in die Rechnerbenutzung und eventuell andere Veranstaltungen 2SWS

Ergänzungsfach: maximal 1 unbenoteter Leistungsnachweis, evtl. Teilnahmebestätigungen, Prüfung besteht aus bis zu 2 Prüfungsleistungen (mündlichen Prüfungen oder Klausuren)

2.2 Lehrveranstaltungszyklen

2.2.1 Mathematische Grundlagen der Informatik

Der Zyklus *Mathematische Grundlagen der Informatik* stellt das für das Informatikstudium nötige mathematische Grundwissen bereit. Zum Erwerb grundlegender mathematischer Fähigkeiten ist das regelmäßige, selbständige Lösen von Übungsaufgaben von besonderer Wichtigkeit. Dementsprechend kommt im Zyklus Mathematische Grundlagen der Informatik den Übungen eine besondere Bedeutung zu.

In der Vorlesung M1 werden zum einen mathematische Grundbegriffe eingeführt. Zum anderen werden besonders informatiknahe Themen aus dem Gebiet der *Diskreten Mathematik* behandelt, wie zum Beispiel elementare Kombinatorik, erzeugende Funktionen, Graphen. Der Teil M2 ist der *Analysis* gewidmet, wobei Differential- und Integralrechnung einer Veränderlichen im Zentrum der Betrachtungen stehen. In M3 wird die *Lineare Algebra* von den Grundbegriffen bis hin zu Euklidischen Vektorräumen und Anfängen der Eigenwerttheorie behandelt. Der Teil M4 über Stochastik enthält sowohl die grundlegenden Begriffe und Modelle der Wahrscheinlichkeitstheorie als auch eine Einführung in die informatikrelevanten Anwendungsgebiete Statistik, Bedienungsmodelle und Simulation.

Übergeordnete Lernziele des Zyklus sind:

- Erwerb von mathematischem Grundwissen,
- Sicherheit im Umgang mit den für die Informatik wichtigsten mathematischen Begriffsbildungen,
- Beherrschung grundlegender mathematischer Techniken und Methoden, die in der Informatik Anwendung finden,
- Vertrautheit mit der axiomatisch-deduktiven Methode,
- Erlernen mathematisch präzisen Argumentierens, Einüben grundlegender Beweistechniken,
- Fähigkeit zur präzisen Beschreibung mathematischer Sachverhalte (schriftlich und mündlich),
- Umsetzung realer Sachverhalte und Vorgängen in mathematische Modelle .

M1: Diskrete Mathematik

- *Mengen, Abbildungen und Boolesche Algebra*: Verknüpfungen von Aussagen bzw. Mengen, Potenzmenge, injektive, surjektive und bijektive Abbildungen, Aussagenlogik, Boolesche Algebra
- *Zahlen*: natürliche, ganze, rationale, reelle und komplexe Zahlen, elementare Zahlentheorie, Rechnen mit dem Summenzeichen, elementare Kombinatorik, Beweisverfahren
- *Relationen und Funktionen*: Eigenschaften von Relationen, Partitionen und Äquivalenzrelationen, Ordnungsrelationen, Funktionen
- *Algebraische Strukturen*: Gruppen, Ringe, Körper
- *Graphen*: Eulersche Linien, Hamiltonsche Kreise, Färbungsprobleme, gerichtete Graphen, Bäume
- *Erzeugende Funktionen*: Potenzreihen, Konvolutionsprodukt, Lösung von Rekursionen

M2: Analysis

- *Konvergenz und Stetigkeit*: Axiome der reellen Zahlen, Ungleichungen und Betrag, Konvergenz von Folgen, Grenzwerte von Funktionen, Stetigkeit

- *Differentialrechnung*: Ableitung von Funktionen einer Veränderlichen, Ableitungsregeln, Differentiation elementarer Funktionen, Extremstellen und Kurvendiskussion, Regeln von de l'Hospital, Newtonsches Verfahren
- *Trigonometrische und Hyperbolische Funktionen*: Definition und Ableitung der trigonometrischen Funktionen, Umkehrfunktionen der trigonometrischen Funktionen, hyperbolische Funktionen
- *Integralrechnung* : Riemannsches Integral, Fundamentalsatz der Differential- und Integralrechnung, Integrationstechniken, Interpolation, numerische Integration, uneigentliche Integrale
- *Reihen*: Konvergenzkriterien, Potenzreihen, Taylorscher Satz, Taylorreihen, Fourierreihen
- *Funktionen mehrerer Variablen*: Stetigkeit, partielle Ableitungen, Definition und Berechnung zweidimensionaler Integrale
- *Differentialgleichungen*: Lineare Differentialgleichungen erster Ordnung, Methode des Koeffizientenvergleichs, Variation der Konstanten, komplexe Exponentialfunktion, lineare Differentialgleichungen zweiter Ordnung mit konstanten Koeffizienten, getrennte Veränderliche

M3: Lineare Algebra

- *Vektorräume*: reelle und komplexe Vektorräume, Vektorräume über beliebigen Körpern, Untervektorräume, lineare Unabhängigkeit, Dimension, Basis
- *Lineare Abbildungen*: Kern und Bild einer linearen Abbildung, lineare Abbildungen und Matrizen, Drehungen und Spiegelungen
- *Matrizenrechnung*: Multiplikation von Matrizen, Rang einer Matrix, elementare Umformungen, Inversion von Matrizen
- *Die Determinante*: Definition und Berechnung von Determinanten, Determinantenformel für die inverse Matrix, Leibnizsche Formel
- *Lineare Gleichungssysteme*: Cramersche Regel, Gaußscher Algorithmus, Numerik linearer Gleichungssysteme
- *Euklidische Vektorräume*: Skalarprodukte, orthogonale Vektoren, Erhard Schmidtsches Orthonormalisierungsverfahren, orthogonale Abbildungen, Hessesche Matrix und Extrema von Funktionen mehrerer Veränderlicher
- *Eigenwerte*: Eigenwerte und Eigenvektoren, charakteristisches Polynom, Eigenräume, geometrische und algebraische Vielfachheit

M4: Stochastik

- *Wahrscheinlichkeits-Modelle*: Merkmale, Ereignisse, Wahrscheinlichkeiten, Dichten, Exponential- und Normalverteilung, Verteilungsfunktionen
- *Mehrstufige Modelle*: Bedingte Wahrscheinlichkeiten, mehrstufige Versuche, Unabhängigkeit, Ziehungsmodelle, Folgen von gekoppelten Modellen
- *Bildmodelle*: Zufallsvariable, induzierte Modelle, Binomial-, Poisson-, hypergeometrische und geometrische Verteilung, Randverteilungen, unabhängige Zufallsvariable, Faltung
- *Kenngrößen*: Median, Quantile, Erwartungswert, Streuung, Varianz, Korrelation, mehrdimensionale Normalverteilung, zufällige Summen
- *Statistik*: Schätzen, Intervallschätzung, Hypothesen-Tests, Anpassungs- und Unabhängigkeitstests, lineare Regression, Varianzanalyse

- *Bedienungsmodelle*: Markov-Ketten, einfache Bedienungsmodelle, Gleichgewichtsverteilungen, Leistungsmaße, einfache Bedienungsnetze
- *Simulation*: Erzeugen gleichverteilter Zufallszahlen, Zufallszahlen mit anderen Verteilungen, Simulationsverfahren, Simulation stochastischer Prozesse

2.2.2 Formale Grundlagen der Informatik

Der Zyklus *Formale Grundlagen der Informatik* beschäftigt sich auf mathematischer Basis mit Abstraktionen, Modellbildungen und Verfahren zur Beschreibung und Analyse von Algorithmen und Prozessen. Formale Methoden spielen in der Informatik die Rolle eines "Denkzeugs", mit dem der (abstrakte) Kern einer Sache knapp und präzise beschrieben werden kann. Erst auf der Basis eines sauberen theoretischen Fundaments wird es möglich, solche Beschreibungen zu formulieren und deren Analysen vorzunehmen. Auf Bezüge zur Anwendung dieser Konzepte, wie sie insbesondere in anderen Veranstaltungen des Grundstudiums vorkommen, wird Wert gelegt.

Die *Logik* bildet die Grundlage für eine formale Semantik von sprachlichen Beschreibungen und Anweisungen in Programmier-, Spezifikations- und Repräsentationssprachen. Das Gebiet *Automatentheorie* behandelt einfache mathematische Modelle, die dem Computer und Algorithmen zugrunde liegen. In dem Teilgebiet *Formalen Sprachen* wird der prinzipielle, strukturelle Aufbau von Programmier- und Spezifikationssprachen untersucht. Die Theorie der *Berechenbarkeit* untersucht die Abgrenzung zwischen effektiv Ausführbarem und prinzipiell niemals Möglichem. Die *Komplexitätstheorie* fragt danach, welchen rechnerischen Aufwand die Lösung gewisser Problem erfordert. *Phänomene der Nebenläufigkeit* treten bei parallelen Rechnerarchitekturen, höheren Programmiersprachen und in Rechnernetzen auf. Ihr korrekter und effizienter Einsatz erfordert besondere Konzepte und Verfahren.

Übergeordnete Lernziele des Zyklus sind:

- Methoden der (strukturellen) Mathematik auf informatische Probleme anwenden zu können,
- Definitionsprinzipien und Beweistechniken erlernen und in unterschiedlichen Bereichen anwenden zu können,
- formale Konzepte der Modellierung kennen- und einsetzen zu lernen,
- Determinismus und Nicht-Determinismus als fundamentale und nützliche Begriffe zu verstehen,
- die Grenzen der prinzipiellen Möglichkeiten der Formalisierung mithilfe von theoretisch fundierten Aussagen benennen und einschätzen zu können.

F 1: Logik

- *Einführung in die Aussagenlogik*: wohlgeformte Ausdrücke, Wahrheitswerte, Wahrheitstabellen, Belegungsfunktion.
- *Grundkonzepte der Semantik*: Allgemeingültigkeit, Unerfüllbarkeit, Erfüllbarkeit und Folgerung.
- *Beweistheorie und Ableitungssysteme für die Aussagenlogik*: Schlußregeln, modus ponens, Beziehung zwischen Semantik und Beweisverfahren, Vollständigkeit und Korrektheit von Beweisverfahren.
- *Mechanisierung des aussagenlogischen Beweisens*: Normalformen, Literale und Klauseln, Resolutionsverfahren, Hornklauseln.
- *Syntax der Prädikatenlogik erster Stufe*: Charakterisierung wohlgeformter Ausdrücke, Quantoren und Quantorenkopie, freie und gebundene Variablen, geschlossene Formeln.

- *Semantik der Prädikatenlogik*: Modell, Interpretation, Allgemeingültigkeit, Unerfüllbarkeit, Erfüllbarkeit, Folgerung.
- *Ableitungssysteme und mechanisches Beweisen in der Prädikatenlogik*: Normalformen, Skolemisierung, Resolutionsverfahren der Prädikatenlogik, Unifikation, allgemeinsten Unifikator, Resolutionsverfahren, Beziehung zwischen Semantik und Ableitungssystemen.

F2: Automaten und Kalküle

- *Definitions- und Spezifikationsmethoden*: Induktive und rekursive Definitionen über Mengen, Relationen, Funktionen, formale Sprachen, Operatoren, transitive Hüllen, Graphen und Bäume.
- *Beweismethoden und -techniken*: Schubfachprinzip, Beweis durch Gegenbeispiel, indirekter Beweis, Induktionsprinzip, Diagonalisierung, strukturelle Induktion.
- *Formale Syntax von Programmiersprachen*: Rationale Ausdrücke, Syntaxdiagramme, reguläre und kontextfreie Chomsky-Grammatiken, Ableitungsbegriff, Ableitungs- und Syntaxbaum, Eigenschaften kontextfreier Sprachen, Grenzen der Ausdrucksfähigkeit.
- *Grundlagen kontextfreier Syntaxanalyse*: endlicher Transduktor, Syntaxanalyse für deterministische und nichtdeterministische Grammatiken, top-down und bottom-up Verfahren.
- *Modelle einfacher Automaten*: endlicher Automat, Kellerautomat, Determinismus versus Nichtdeterminismus.
- *Kalküle*: Signaturen, Strukturbegriff, Ableitung und Deduktion, Terme, Termkonstruktion, Beispiele von Kalkülen, Lambda-Kalkül.

F 3: Berechenbarkeit und Komplexität

- *Konkrete Komplexitätstheorie*: Grundlegende Entwurfstechniken (z.B. Teile und Herrsche, Greedy-Methode, rekursive Algorithmen, dynamische Programmierung).
- *Analyse von Datenstrukturen*: Listen, Wörterbücher, Warteschlangen, Suchbäume.
- *Analyse konkreter Algorithmen*: Suchen, Sortieren, transitive Hüllen, kürzeste Wege, spannende Bäume.
- *Entscheidbarkeit und Berechenbarkeit*: primitiv rekursive und allgemein-rekursive Funktionen, Register- und Turing-Maschine als allgemeine Algorithmusdefinition, Halteproblem.
- *Strukturelle Komplexitätstheorie*: Darstellung von Problemen als Mengen, Zeit- und Platzkomplexitätsklassen, Landausche Notation, Beschleunigungs- und Kompressions-Sätze, Reduktionsbegriff, NP-Vollständigkeit.
- *Höhere Grammatiken*: Chomsky Typ-0 und Typ-1 Grammatiken, Normalformen, Chomsky-Hierarchie.

F 4 - Parallelität und Nebenläufigkeit

- *Transitionssysteme*: Zustand, Aktion, Erreichbarkeit, Semantik der Nebenläufigkeit, Verklemmungsfreiheit und Fairneß, Prinzipien der Spezifikation, Lebendigkeit und Sicherheit, erste Begegnung mit formalen Modellen (Prozessalgebra, Petrinetze).
- *Nichtsequentielle Programme*: Modellierung, Synchronisations- und Kommunikationsprinzipien, Speicher- und Rendezvous-Synchronisation, klassische Fallbeispiele, Serialisierbarkeit und Funktionalität, Invarianten.

- *Petrinetze und automatenbasierte Modelle*: Netzklassen, Prozesse, Verifikationstechniken mit linearer Algebra, strukturelle Analysemethoden.
- *Parallele Algorithmen*: synchrone parallele Algorithmen, Zeit- und Prozessor-Komplexität.
- *Verteilte Algorithmen*: Grundannahmen und Darstellungsformen, wichtige Problemklassen, Auswahlalgorithmen, verteilter wechselseitiger Ausschluss.
- *Systeme von Funktionseinheiten*: Auftrag, Funktionseinheit, Durchsatz, Füllung, Gesetz von Little, Sättigungsfüllung und Verkehrsengepass, Aufträge in Systemen von Funktionseinheiten, Modellierung von realen Prozessen in Organisationen und in der Fertigung.

2.2.3 Technische Informatik

Ziel der Ausbildung des Zyklus *Technische Informatik* ist die Vermittlung eines begrenzten und wohlausgewählten Theorie- und Methodenrepertoires für den Entwurf, die Realisierung, die Konfigurierung und die angemessene Nutzung von komplexen Rechen- und Kommunikationssystemen sowie ihrer Basiskomponenten (u. a. unter Berücksichtigung technologischer, ökonomischer und anwendungsspezifischer Randbedingungen). Der Lehrstoff wird durch Realwelt-Beispiele ergänzt und ermuntert die Studierenden zu eigenständiger Beschäftigung mit den Inhalten. Schwerpunkt der Ausbildung in diesem Zyklus liegt auf Systemen, die letztlich als physisch-materiell verwirklicht zu denken sind.

Dabei bilden die Teile T1 und T2 einerseits sowie die Teile T3 und T4 andererseits jeweils einen in sich zusammenhängenden - durch besondere Methoden charakterisierten - Komplex.

Übergeordnete Lernziele des Zyklus sind:

- Verstehen der grundlegenden Konzepte, Organisationsformen und Entwurfsmethoden der Rechnerarchitektur und der Vernetzung einschließlich der Betriebssoftware,
- Verstehen der Prinzipien der technischen Realisierung von Rechner- und Kommunikationshardware,
- Kennenlernen von Methoden zur formalen Beschreibung, Modellierung und Bewertung von Rechen- und Kommunikationssystemen auf unterschiedlichen Abstraktionsebenen,
- Erwerben der Fähigkeit zur Nutzung hardwarenaher Schnittstellen.

T1: Digitale Systeme 1

- *Grundbegriffe der Informationsverarbeitung*: Information, Repräsentation, Kommunikation, Zahlensysteme, System-Begriff, Codierung
- *Arithmetiken*: Ganzzahl- und Gleitkomma-Arithmetik
- *Hardwarebeschreibungssprachen/Entwurfsebenen*: Verhaltens- und Strukturbeschreibung
- *Schaltnetze*: Boolesche Algebra, Schaltnetzsynthese, Zeitverhalten
- *Schaltwerke*: Moore- und Mealy-Modell, Entwurf von getakteten Schaltwerken, Flipflops
- *Ein einfacher von-Neumann-Rechner*: Architektur, Befehlssatz, Assemblierung

T2: Digitale Systeme 2

- *Elektrotechnische Grundlagen*: lineare Netze
- *Nichtlineare Bauelemente*: Halbleiterdiode, Transistoren
- *Digitale Schaltungstechnik*: Logik-Glieder, Bus-Systeme
- *Strukturen von Halbleiterspeichern*: Arten der Adressierung, Realisierung statischer und dynamischer Speicher

- *Programmierbare Bausteine*: Arten der Programmierung, interne Struktur, Anwendung
- *Herstellung integrierter Schaltungen*: Lithographie, chemische und physikalische Prozeß-Schritte
- *Wirtschaftliche Aspekte der Hochintegration*: Fehlermechanismen und Chipausbeute, Entwurfskomplexität und Entwurfsstile, Test gefertigter Schaltungen

T3: Rechnerorganisation und Betriebssysteme

- *Grundzüge der Rechnerarchitektur*: Entwurfsprinzipien, Befehlssätze, Adressierung, Speicherhierarchie
- *Prozessorimplementation*: Beispiel eines modernen Prozessors, Pipelining, Parallelisierung und weitere effizienzsteigernde Maßnahmen
- *Ein- und Ausgabe*: Peripheriegeräte, Schnittstelle zu Prozessoren, Bussysteme, Interrupt-Behandlung
- *Grundelemente von Betriebssystemen*: Prozesse, Interprozeßkommunikation, Betriebsmittelverwaltung, Dateisysteme
- *Prozeßscheduling*: Ziele, klassische Verfahren, Termin-Scheduling, Bewertung
- *Schutzmechanismen*: Schutzmodelle, Zugangskontrolle, kryptographische Ansätze

T4: Parallelverarbeitung und Kommunikationsnetze

- *Parallelrechner*: Grundlagen der Parallelverarbeitung, Architekturkonzepte und Realisierungen
- *Informationstheorie und Codierungstheorie*: Informationsgehalt, Redundanz, fehlererkennende und fehlerkorrigierende Codes
- *Rechnernetzarchitekturkonzepte und Referenzmodelle*: u.a. Dienst, Protokoll, Schichtenstruktur, Adressierung, Standardarchitekturen, Internet-Architektur
- *Übertragungstechnik und Datenübertragung*: physikalische Übertragungswege, Signalübertragung, gesicherte Datenübertragung über Punkt-zu-Punkt-Verbindungen
- *Vermittlungsnetze*: Vermittlungstechniken, Weegermittlung, Flußkontrolle, Internet-Protokolle
- *Lokale Rechnernetze*: Zugriffskontrolle in Bussystemen und Ringnetzen, LAN-Standards
- *Modellierungs-, Analyse- und Entwurfsverfahren*: Überblick über Ziele und Verfahren zur Planung, Bewertung und zum Management von Netzen
- *Unterstützung verteilter Anwendungen*: Transportdienste und Transportsysteme, anwendungsorientierte Dienst- und Verwaltungskonzepte

2.2.4 Praktische Informatik

Der Zyklus *Praktische Informatik* behandelt die konzeptionellen und methodischen Grundlagen für die erfolgreiche Entwicklung von komplexen und zuverlässigen Softwaresystemen. Vermittelt werden die zentralen Konzepte der Softwarekonstruktion sowie ihre Anwendung in der praktischen Programmierung. Die Darstellung erfolgt in exemplarischer Form anhand der Programmiersprachen Prolog, Scheme und Java als typischen Repräsentanten der drei grundlegenden Paradigmen: deklarative, applikative und zustandsorientierte Programmierung. Damit werden die individuellen Voraussetzungen zum Erlernen beliebiger Programmiersprachen geschaffen.

Fragen der Programmiermethodik stellen ein verbindendes Element zwischen den verschiedenen Themenkomplexen dar. Durch umfangreiche Übungen, sowie eine vertiefte Behandlung des

zustandsorientierten Paradigmas im zweiten und dritten Semester soll die Fähigkeit zur praktischen Anwendung der verschiedenen Abstraktionstechniken bei der Lösung komplexer Softwareentwicklungsaufgaben herausgebildet werden. Ein besonderer Schwerpunkt ist hierbei mit den Methoden der objektorientierten Modellierung gegeben. Behandelt werden ferner auch Fragen der nebenläufigen und verteilten Programmierung.

Übergeordnete Lernziele des Zyklus sind:

- Kennenlernen von grundlegenden Konzepten, Methoden und Werkzeugen der Softwareentwicklung
- Erwerben der Fähigkeit zur kritischen Auswahl von Methoden und Werkzeugen
- Erwerben der Fähigkeit zum selbständigen und systematischen Lösen programmiertechnischer Aufgaben

P1: Deklarative und applikative Programmierung

- *Programmieren mit Relationen*: elementare und komplexe Terme, extensionale Spezifikation (Fakten), Anfragen, Unifikation, Unterspezifikation, Suche, Nichtdeterminismus, komplexe Anfragen, intensionale Spezifikation (Regeln), spezielle Relationstypen
- *Eingebettete Terme*: Listen, Bäume und Graphen als Terme, Rekursion über Termstrukturen (Listenverarbeitung, Peano-Arithmetik, Baumtraversieren)
- *Grenzen der deklarativen Programmierung*: Arithmetik, applikative Auswertungsumgebung
- *Grundbegriffe der applikativen Programmierung*: applikatives Verarbeitungsmodell, Namensabstraktion, Terme, Bindungsumgebungen, funktionale Abstraktion
- *Funktionale Applikation*: Werte, Reduktionsstrategien, Striktheit, Bezugstransparenz, Umgebungsmodell
- *Rekursive Funktionen und Algorithmen*: Lineare Rekursion und Listen, Aufwand rekursiver Prozesse, Endrekursion, Akkumulatoren, Baumrekursion, indirekte Rekursion, verschachtelte Rekursion
- *Funktionen höherer Ordnung*: Der Baukasten (Abilden, Filtern, Falten), Verknüpfung von Funktionen (Komposition, Curry-Verfahren, closures) funktionales backtracking-Schema, verzögerte Auswertung, Ströme

P2: Zustands- und objektorientierte Programmierung

- *Grundkonzepte der zustandsorientierten Programmierung*: Variablen- und Zustandsbegriff, Anweisung, Objekt und Wert, Ablaufsteuerung, axiomatische Basis der zustandsorientierten Programmierung, operationale Semantik
- *Typkonzept der zustandsorientierten Programmierung*: Typbegriff, einfache und zusammengesetzte Typen, vorgegebene und benutzerdefinierte Typen, Verbund und Reihung, Referenztypen, Prozedurtypen, polymorphe Typsysteme
- *Abstraktionmechanismen*: prozedurale Abstraktion, Datenabstraktion, abstrakte Datentypen
- *Modulkonzept*: Sichtbarkeit und Namensräume, Kapselung, Geheimnisprinzip, Gestaltung von Schnittstellen, Objekt- und Typmodul, Vertragsmodell, Schnittstellen und Zusicherungen, Zusicherungen und Ausnahmebehandlung
- *Konzepte der objektorientierten Modellierung*: Typ und Klasse, Vererbung, objektorientierte Architektur
- *Konzepte und Implementierung von Programmiersprachen*: Namen und Namensräume, Typprüfung, Bindungsmechanismen; Übersetzen, Binden und Laden; Laufzeitsystem, Programmierwerkzeuge und -umgebungen

- *Grundkonzepte der Softwaretechnik*: Spezifikation, Implementation und Validation, Projektmodelle

P3: Algorithmen und Datenstrukturen

- *Dynamische Datenstrukturen*: Spezifikation und Implementation von linearen Listen, Kellern, Schlangen, Bäumen, Graphen; Algorithmen und ihre Bewertung
- *Anwendungen von dynamischen Datenstrukturen*: Behälterklassen, Suchen in geordneten Datenbeständen
- *Grundzüge der Daten- und Wissensmodellierung*: programminterne und persistente Datenstrukturen, Datenbankmodelle (u.a. relational, objektorientiert), semantische Netze
- *Nebenläufige Programmierung*: Prozeßbegriff, Synchronisation, Kommunikation zwischen Prozessen, Prozeß vs. Threads
- *Anwendungen nebenläufiger Programmierung*: Betriebsmittelverwaltung, prozeßorientierte Simulation
- *Verteilte Programmierung*: Programmierschnittstellen, verteilte Systemarchitekturen

2.2.5 Informatik, Mensch und Gesellschaft

Der Zyklus *Informatik-Mensch-Gesellschaft* vermittelt grundlegende Kenntnisse über die Informatik als Wissenschaft, über das ihr zugrundeliegende Paradigma der Informationsverarbeitung sowie über die Stellung von Informatiksystemen in der Gesellschaft. Dabei zeigt sich die Informatik als ein Ergebnis einer jahrtausendelangen Entwicklung der Formalisierung in Wissenschaft und Praxis, die die Grundlagen zur Entwicklung von Informatiksystemen bereitgestellt hat. Die Informatik ist durch eine hohe Vielfalt charakterisiert, die in verschiedenen Sichten der Wissenschaft, in mannigfaltigen Technologien und Anwendungen sowie in verschieden ausgeprägten fachübergreifenden Verflechtungen zum Ausdruck kommt.

Grundlegende Kenntnisse der natürlichen Informationsverarbeitung werden benötigt, um Ähnlichkeiten und Unterschiede zwischen Mensch und Computer zu verstehen, die Mensch-Computer-Interaktion zu gestalten, oder menschliche Fähigkeiten mit Informatikmethoden zu simulieren.

Informatiksysteme bringen häufig tiefgreifende Veränderungen für das Individuum, die Arbeitsorganisation und die Gesellschaft. Sie werden interessengeleitet entwickelt und eingesetzt. Informatiker und Informatikerinnen sind direkt verantwortlich für die von ihnen entwickelten technischen Lösungen. Zugleich sind sie zusammen mit anderen Akteuren an einem gesellschaftlichen Veränderungsprozeß beteiligt, für den sie Verständnisgrundlagen erarbeiten müssen, um Gestaltungsoptionen verantwortungsvoll wahrnehmen zu können.

Übergeordnete Lernziele des Zyklus sind:

- die Verankerung der Informatik im soziokulturellen Hintergrund und der Geistesgeschichte zu verstehen,
- die Wechselwirkungen zwischen informatischer Modellierung und der Wirklichkeit einschätzen zu können,
- wissenschaftliche Methoden zum Verständnis von Informatiksystemen im Kontext kennen zu lernen,
- die Rolle von Informatikern und Informatikerinnen im Berufsleben zu reflektieren.

IMG1: Informatik, eine Standortbestimmung

- *Informatisch handeln*: Fallstudie; Berufsfelder, Rolle von Informatikerinnen / Informatikern
- *Hintergrund der Informatik*: Rechnen und Formalisierung, formalisierte Praxis, Maschinen und Automaten
- *Informationsverarbeitung als Leitbild*: Informationsbegriff(e) der Informatik, Information und Datenverarbeitung, Begriff Informatiksystem, Information und Kommunikation
- *Informatische Modellierung*: Operationale Modelle, Operationale (Re-)Konstruktion, Modellierung als sozialer Prozeß, Modell und Wirklichkeit
- *Informatiksysteme als Wirkungsinstanzen*: Wirkungsweisen von Informatiksystemen, Problemfelder Beherrschbarkeit und Fehlerhaftigkeit, Zusammenwirken von Mensch und Informatiksystem, gesellschaftliche Veränderungen durch Informatiksysteme
- *Informatik als Wissenschaft*: Sozialgeschichte der Informatik; Sichten der Informatik als Kybernetik, Computer Science, Computing Science, Kognitionswissenschaft, Informationswissenschaft, Strukturwissenschaft, Ingenieurwissenschaft, Gestaltungswissenschaft
- *Ausdifferenzierung der Informatik*: exemplarisch anhand der Hamburger Studienprofile

IMG3: Grundlagen der natürlichen Informationsverarbeitung

- *Biologische, insbesondere neurowissenschaftliche Grundlagen*: Struktur und Funktionsprinzipien des Gehirns, Einführung in die Modellierung neuronaler Systeme
- *Perzeption*: Grundlagen der visuellen, auditiven und taktilen Wahrnehmung; Schwerpunkt visuelle Wahrnehmung (u.a. Gestalt, Form- und Farbwahrnehmung), Perzeptions- / Aktionszyklus
- *Gedächtnis und Repräsentation von Information*: Grundlagen der Gedächtnispsychologie, Arbeitsgedächtnis und Langzeitgedächtnis, modalitätsspezifische Gedächtnisleistungen
- *Denken und Problemlösen*: Problemlösen als Transformation von Repräsentationen, Denken und Gedächtnis
- *Kommunikation*: Sprachliche Kommunikation: natürliche Sprache, nicht-sprachliche Kommunikation: Kommunikation mit Diagrammen, Gestik
- *Lernen*: Lerntheorien, Lerntypen, mentale Modelle
- *Denken und Handeln*: Handlungsregulation, Kognition und Koordination / Kooperation in Gruppen

IMG4: Informatiksysteme in der Gesellschaft

- *Informationstechnik-unterstützte Modellierung und Gestaltung für Organisationen*: Techniknutzung als historischer Prozeß; Technik-, Arbeits- und Organisationskonzepte in Büro und Verwaltung sowie in der Produktion
- *Technikentwicklung in der Gesellschaft*: Theorien zur Technikentwicklung; Akteursmodell der Informatik
- *Technikfolgenabschätzungs-, Technikbewertungs- und Technikgestaltungsforschung (TA)*: Begründung, Schwierigkeiten und Perspektiven; TA-Methoden; TA und Informatik (Ebenen Techniksyste-me und Gesellschaft, Produkt- & Dienstleistungs-TA für Informations- und Kommunikationstechnik sowie für informationstechnik-unterstützte Gestaltung von Organisationen).
- *Gesellschaftliche Wechselwirkungen und Gestaltungsoptionen*: Informationstechnik, Innovationen, Wachstum, Arbeitsmarkt und Umwelt

- *Informationsgesellschaft, Nachhaltige Entwicklung und Informatiksysteme*: Agenda 21; Beiträge der Informations- und Kommunikationstechnik für nachhaltige Entwicklung; Dematerialisierung, Ressourceneffizienz, Rebound-Effekte; Stoffstrommanagement; nachhaltige Szenarien; weltwirtschaftliche Rahmenbedingungen
- *Verantwortung und Leitbild der Informatikerin/ des Informatikers*: Leitbilddoptionen (Ingenieur, Architekt, Anwalt, Moderator, Organisator?); Fragen der Berufsethik
- *Datenschutz, Netze, Sicherheit*: Problemaufriß, rechtliche und technische Optionen, Risiken der Informationstechnik, kritische Systeme

2.3 Anforderungen für Leistungsnachweise und Teilnahmebestätigungen

2.3.1 Übungen

In den Übungen geht es um aktive Mitarbeit bei den Übungsterminen sowie um das Bearbeiten von Aufgaben. Die Aufgaben werden in der Regel in Kleingruppen von 2-3 Studierenden gelöst. Für die Bewertung von Aufgaben ist eine Grundlage durch Punkte bzw. Fallunterscheidungen wie “nicht bearbeitet”, “bearbeitet”, “erfolgreich bearbeitet” anzugeben.

Einen Übungsschein erhält, wer aktiv mitarbeitet, regelmäßig die Aufgaben bearbeitet und insgesamt mindestens 50% der zu vergebenden Punkte erwirbt bzw. die Hälfte der Aufgaben erfolgreich bearbeitet. Die aktive Mitarbeit wird an den Übungsterminen zum Beispiel durch Präsentation der gelösten Aufgaben erbracht. Die genauen Modalitäten der Scheinvergabe werden am Anfang der Übungen bekanntgegeben. In Zweifelsfällen findet am Ende der Übungen eine individuelle, mündliche Rücksprache statt, die sich direkt auf die Übungsaufgaben bezieht.

2.3.2 Proseminare

Proseminare dienen zur Einführung in das wissenschaftliche Arbeiten im Grundstudium. Die Studierenden halten ein Referat über ein wissenschaftliches Thema, erarbeiten dazu unter Anleitung die Literatur und fertigen eine schriftliche Ausarbeitung an. Nach Möglichkeit leiten sie auch die Diskussion und wenden Moderationstechniken an. Dabei ist die Zusammenarbeit in Kleingruppen von 2-3 Studierenden erlaubt. Eine Teilnahmebestätigung erhält, wer eine Proseminarsitzung durch Referat und Diskussion (ggf. in Zusammenarbeit mit anderen) gestaltet, die schriftliche Zusammenfassung zur Sitzung anfertigt und sich regelmäßig aktiv an den Diskussionen im Proseminar beteiligt.

Zur Überprüfung der regelmäßigen Beteiligung kann eine Anwesenheitsliste geführt werden.

2.3.3 Praktika

In den Praktika geht es um die Anwendung oder die Einübung von Fertigkeiten durch die Bearbeitung von Aufgaben (Experimenten oder größeren Programmen). Die Aufgaben werden in der Regel in Kleingruppen von 2-3 Studierenden gelöst. Für die Bewertung von Aufgaben ist eine Grundlage durch Punkte bzw. Fallunterscheidungen wie “nicht bearbeitet”, “bearbeitet”, “erfolgreich erarbeitet” anzugeben.

Einen Praktikumsschein erhält, wer regelmäßig die Aufgaben bearbeitet und insgesamt mindestens 50% der zu vergebenden Punkte erwirbt bzw. die Hälfte der Aufgaben erfolgreich bearbeitet. Zur Bearbeitung von Aufgaben gehört auch die Dokumentation (Protokolle oder Programm-Dokumentation) sowie die Präsentation der Ergebnisse (Vorführung oder Programm-Demonstration) an den Praktikumsterminen. Die genauen Modalitäten der Scheinvergabe werden

am Anfang des Praktikums bekanntgegeben. In Zweifelsfällen findet am Ende des Praktikums eine individuelle, mündliche Rücksprache statt, die sich direkt auf die Praktikumsaufgaben bezieht.

2.4 Diplom-Vorprüfung

Zulassungsvoraussetzungen für die Diplom-Vorprüfung sind:

- die für die Lehrveranstaltungszyklen erforderlichen Übungsscheine: M2 oder M3, M4, F1, T3 oder T4, P2 und P3;
- zwei Scheine für Praktika, davon je eines aus den Gebieten *Praktische* und *Technische Informatik*;
- zwei Teilnahmebestätigungen für Proseminare, davon eines aus dem Gebiet *Informatik, Mensch und Gesellschaft*, eines aus einem anderen Gebiet der Informatik.

Die Diplom-Vorprüfung im Hauptfach besteht aus den Fachprüfungen

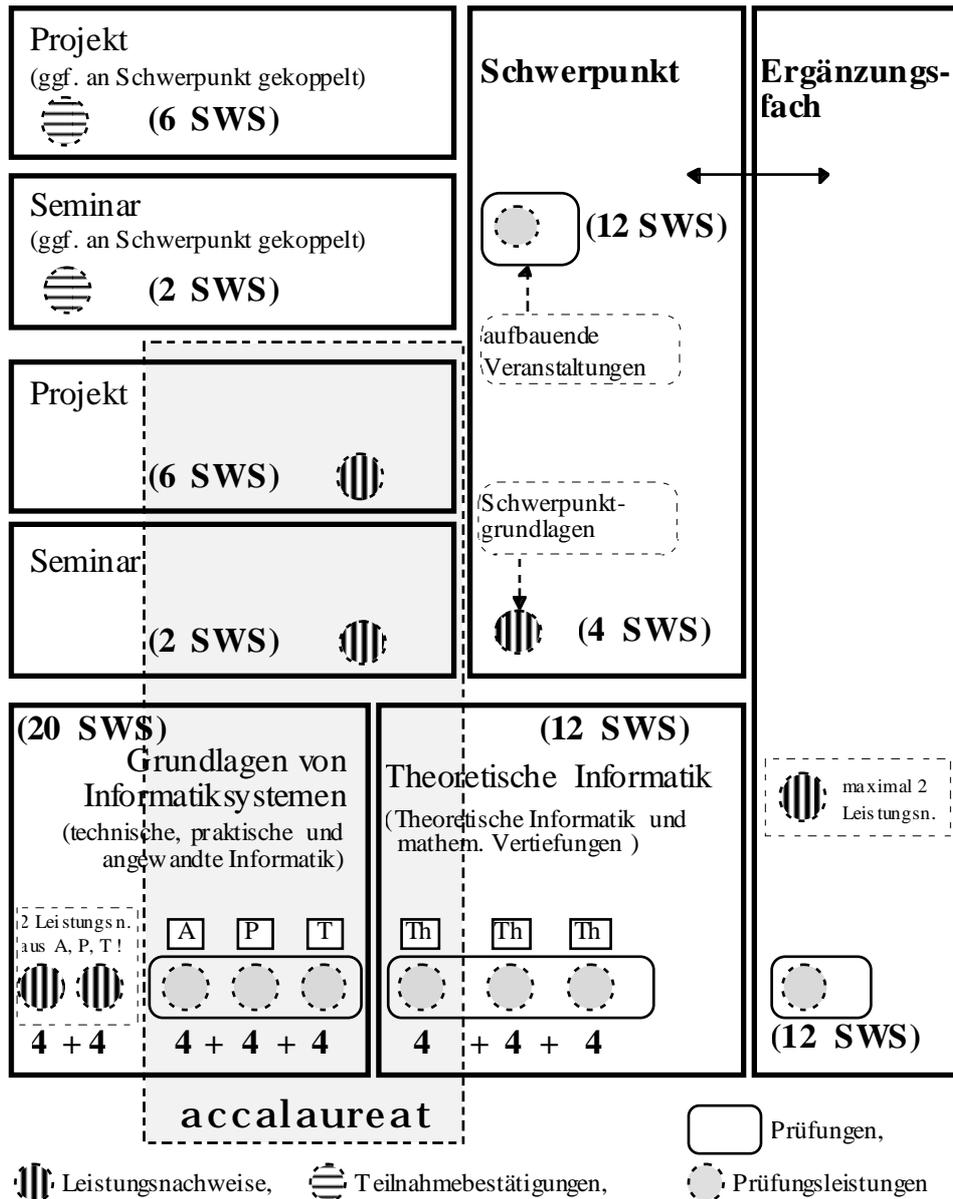
- *Formale Grundlagen der Informatik*: Klausur über F1 und F2, mündliche Prüfung über F3 und F4;
- *Mathematische Grundlagen der Informatik*: Klausur über M1, mündliche Prüfung über M2 und M3;
- *Praktische Informatik*: Klausur über P1, mündliche Prüfung über P2 und P3, mündliche Prüfung über IMG1, IMG3 und IMG4;
- *Technische Informatik*: Klausur über T1 und T2, mündliche Prüfung über T3 und T4.

Die Modalitäten der Ergänzungsfachprüfung ergeben sich aus den Vereinbarungen mit dem zuständigen Fachbereich. Als Zulassungsvoraussetzung können ein unbenoteter Leistungsnachweis (Übungsschein, Praktikumsschein) sowie Teilnahmebestätigungen für Proseminare oder Seminare verlangt werden. Die Prüfung kann aus bis zu 2 Prüfungsleistungen (Klausuren oder mündlichen Prüfungen) bestehen.

3. Hauptstudium

3.1 Übersicht

Das Hauptstudium besteht für alle Studierenden aus den gleichen Bausteinen. Dabei gibt es eine erheblich größere Wahlfreiheit als im Grundstudium, insbesondere keine Pflichtlehrveranstaltungen. Im Baccalaureatsstudiengang kann die Wahlfreiheit nach den hier angegebenen Grundregeln voll ausgeschöpft werden, im Diplomstudiengang wird sie durch das gewählte Studienprofil bestimmt.



3.2 Grundlagenveranstaltungen

Gemäß der Studien- und Prüfungsordnung gibt es Grundlagenveranstaltungen in den Fächern *Grundlagen von Informatiksystemen (GRIS)*, die die Angewandte, Praktische und Technische Informatik umfassen, sowie *Theoretische Informatik*. 5 Veranstaltungen in GRIS (davon 3 mit

mündlicher Prüfung und 2 mit Leistungsnachweis) sowie 3 Veranstaltungen in Theoretischer Informatik (mit je einer mündlichen Prüfung) sind zu absolvieren.

Die *Wahlmöglichkeiten* auf der Ebene der Grundlagenveranstaltungen werden durch die *Studienprofile* bestimmt. In fast allen Schwerpunkten treten bestimmte Grundlagenveranstaltungen des GRIS-Katalogs als “Schwerpunktgrundlagen” auf, für sie ist ein Leistungsnachweis erforderlich.

Folgende Kataloge von Veranstaltungen sind dafür maßgeblich:

Grundlagen von Informatiksystemen

Angewandte Informatik:

- Computergraphik und Bildsynthese (CGB),
- Diskrete Optimierung und Simulation (DOS),
- Gestaltbarkeit und Beherrschbarkeit von Informatiksystemen (GBI)

Praktische Informatik:

- Datenbanken und Informationssysteme (DIS),
- Verteilte Softwaresysteme (VSS),
- Wissensbasierte Systeme (WBS)
- Soft Computing und Neuronale Netze (SNN)
- Softwaretechnik und Softwareergonomie (STE),

Technische Informatik:

- Eingebettete Systeme (EBS),
- Datenkommunikation und Rechnernetze (DKR),
- Rechnerarchitektur und Mikrosystemtechnik (RAM)

Theoretische Informatik:

- Automaten und Komplexität (AUK),
- Logik und Semantik (LOS),
- Prozesse und Nebenläufigkeit (PNL),
- Systemtheorie (STH)

3.3 Festlegungen zu Seminaren, Projekten und Projektseminaren

Seminare im Umfang von 2 SWS dienen der Erörterung ausgewählter wissenschaftlicher Probleme im Hauptstudium. Die Studierenden werden in der Arbeit nach wissenschaftlichen Grundsätzen und der Darstellung wissenschaftlicher Inhalte geschult. Sie erarbeiten selbständig die benötigte Literatur, gestalten einen Seminartermin durch Vortrag und Diskussion und liefern dazu eine schriftliche Zusammenfassung. Dabei ist die Zusammenarbeit in Kleingruppen von 2-3 Studierenden erlaubt. Eine Teilnahmebestätigung erhält, wer diese Leistungen erbringt und sich regelmäßig aktiv an den Diskussionen im Seminar beteiligt. Für einen Seminarschein ist darüber hinaus die Anfertigung einer Seminararbeit erforderlich. Zur Überprüfung der regelmäßigen Beteiligung kann eine Anwesenheitsliste geführt werden.

Projekte im Umfang von 6 SWS dienen der Bearbeitung größerer theoretischer, konstruktiver oder experimenteller Aufgaben und können in zwei aufeinander folgenden Semestern stattfinden. Ein Projekt kann in Zusammenarbeit mit einem Forschungsprojekt des Fachbereichs oder mit einer Einrichtung außerhalb des Fachbereichs veranstaltet werden. Wegen des besonderen Arbeitsaufwands wird den Studierenden empfohlen, nicht mehr als ein Projekt zur gleichen Zeit zu belegen.

Projekte werden in Gruppen durchgeführt und ermöglichen das Erlernen von Gruppenarbeit. Die gesamte Projektgruppe arbeitet auf ein gemeinsames Ergebnis hin. Dabei kann eine Projektgruppe in Kleingruppen von 2-3 Studierenden gegliedert werden.

Eine Teilnahmebestätigung erhält, wer kontinuierlich an den Projektterminen teilnimmt und sich in die verschiedenen Aufgaben der Gruppenarbeit einbringt. Zur Projektarbeit gehört auch die Dokumentation (Protokolle oder Programm-Dokumentation) sowie die öffentliche Präsentation der Ergebnisse (Vorführung oder Programm-Demonstration) an den Projektterminen.

Einen Projektschein erhält, wer darüber hinaus einen Projektbericht erstellt, der die Planung des Vorgehens, dessen Verlauf und die Durchführung sowie die erreichten Ergebnisse zusammenfaßt. Die genauen Modalitäten der Scheinvergabe werden am Anfang des Projekts bekanntgegeben. In Zweifelsfällen findet am Ende des Projekts eine individuelle, mündliche Rücksprache statt, die sich direkt auf die Projektarbeit bezieht.

Soll der Projektbericht als Baccalaureatsarbeit anerkannt werden, so muss dies vor Beginn des Projektes beim Prüfungsausschuss beantragt werden.

Projektseminare im Umfang von 4SWS integrieren thematisch einen Seminarteil mit einem Projektteil von je etwa 2 SWS. Der Seminarteil entspricht im wesentlichen einem eigenständigen Seminar. Der Projektteil entspricht weitgehend einem eigenständigen Projekt, ist allerdings auf ein Semester beschränkt.

Eine Teilnahmebestätigung erhält, wer die dafür geforderten Leistungen in Analogie zu Seminaren bzw. Projekten erbringt. Für Projektseminare werden nur Teilnahmebestätigungen vergeben. So kann ein Paar Seminar / Projekt durch zwei Projektseminare ersetzt werden.

3.4 Baccalaureatsprüfung und Diplomprüfung

3.4.1 Baccalaureatsprüfung

Die Zulassungsvoraussetzungen für das Baccalaureat sind

- ein Seminarschein,
- die Baccalaureatsarbeit, die einem ausführlichen Projektbericht entspricht.

Die Baccalaureatsprüfung besteht aus 4 mündlichen Prüfungen zu Grundlagenveranstaltungen, je eine davon zu Angewandter, Praktischer, Technischer und Theoretischer Informatik.

3.4.2 Diplomprüfung

Die Zulassungsvoraussetzungen zur Diplomprüfung sind:

- Leistungsnachweise über zwei nicht zur Fachprüfung *Grundlagen von Informatiksysteme* ausgewählte Grundlagenveranstaltungen aus den Gebieten Angewandte, Praktische und Technische Informatik;

- der Leistungsnachweis über die Schwerpunkt-Grundlagen;
- ein Seminar- und ein Projektschein (bzw. die entsprechenden Projektseminarscheine), sie sind vor Beginn der Diplomarbeit vorzulegen;
- Teilnahmebestätigungen für ein weiteres Seminar und ein weiteres Projekt (bzw. die entsprechenden Projektseminare).

Die Diplomprüfung besteht aus der Diplomarbeit sowie den Fachprüfungen Grundlagen von Informatiksystemen, Theoretische Informatik, Schwerpunkt und Ergänzungsfachprüfung.

Die Fachprüfungen in Informatik werden erbracht durch

- *Grundlagen von Informatiksystemen*: 3 mündliche Prüfungen zu Grundlagenveranstaltungen, je eine davon zu Angewandter, Praktischer und Technischer Informatik.
- *Theoretische Informatik*: 3 mündliche Prüfungen zu Grundlagenveranstaltungen aus der Theoretischen Informatik;
- *Schwerpunkt*: eine mündliche Prüfung über die aufbauenden Veranstaltungen des Schwerpunktes.

Die Modalitäten für die Diplomprüfung im Ergänzungsfach ergeben sich aus den Vereinbarungen mit dem zuständigen Fachbereich.

4 Studienprofile

Zu jedem *angebotenen* Studienprofil werden im folgenden die maßgeblichen Grundlagenveranstaltungen, die Schwerpunkte sowie die empfohlenen Ergänzungsfächer angegeben.

Für die empfohlenen Ergänzungsfächer wird ein Rahmenstudienplan entwickelt, der im Hauptstudium profilspezifisch sein kann und den inhaltlichen Bezug zwischen Hauptfach und Ergänzungsfach sicherstellt.

Darüber hinaus können weitere Ergänzungsfächer beantragt werden. Dazu ist ein individueller Studienplan sowie eine Begründung erforderlich, die den inhaltlichen Bezug zum Profil herstellt.

Besteht ein Wunsch nach weitgehend freier Kombination der Bausteine des Hauptstudiums, so ist ein *individuelles* Studienprofil zu beantragen.

Das gewählte (angebotene oder individuelle) Studienprofil wird zu Beginn des Hauptstudiums durch Antrag beim Prüfungsausschuß festgelegt.

4.1 Informatiksysteme in Organisationen

Grundlagenveranstaltungen

Grundlagen von Informatiksystemen

Angewandte Informatik:	GBI
Praktische Informatik:	STE
Technische Informatik:	1 aus DKR, EBS
Nach Wahl:	2 aus CGB, DIS, DOS, VSS, WBS

Theoretische Informatik

LOS, PNL, 1 aus AUK und STH

Schwerpunkte

- Informatiksysteme für Nachhaltige Entwicklung
- Organisationsbezogene Softwareentwicklung

Empfohlene Ergänzungsfächer

Betriebswirtschaftslehre, Biologie, Geographie, Mathematik, Medizin, Meteorologie, Ozeanographie, Philosophie, Politische Wissenschaften, Psychologie, Rechtswissenschaften, Soziologie, Volkswirtschaftslehre.

4.2 Intelligente Systeme

Grundlagenveranstaltungen

Grundlagen von Informatiksystemen:

Angewandte Informatik:	CGB
Praktische Informatik:	WBS
Technische Informatik:	EBS
Nach Wahl:	2 aus SNN, DIS, STE, DOS und DKR

Theoretische Informatik:

LOS, STH, 1 aus AUK und PNL

Schwerpunkte:

- Bildverarbeitung
- Sprachverarbeitung
- Wissensverarbeitung

Empfohlene Ergänzungsfächer

Alle Studiengänge der Sprachwissenschaften, Betriebswirtschaftslehre, Biologie, Elektrotechnik (TUHH), Erziehungswissenschaft, Geographie, Mathematik, Medizin, Meteorologie, Ozeanographie, Philosophie, Physik, Psychologie, Visuelle Kommunikation (HfBK).

4.3 Softwaresystemtechnik

Grundlagenveranstaltungen

Grundlagen von Informatiksystemen

Angewandte Informatik:	GBI
Praktische Informatik:	2 aus DIS, STE, VSS
Technische Informatik:	DKR
Nach Wahl:	1 aus CGB, DOS, RAM, WBS

Theoretische Informatik

LOS, PNL, 1 aus AUK und STH

Schwerpunkte

- Verteilte und Informationssysteme
- Softwareentwicklungsmethodik

Empfohlene Ergänzungsfächer

Betriebswirtschaftslehre, Deutsche Sprache und Literatur, Erziehungswissenschaft, Geographie, Mathematik, Medizin, Meteorologie, Ozeanographie, Philosophie, Physik, Psychologie, Visuelle Kommunikation (HfBK).

4.4 Technikorientierte Informatiksysteme

Grundlagenveranstaltungen

Grundlagen von Informatiksystemen

Angewandte Informatik:	CGB oder DOS
Praktische Informatik:	SNN oder DIS oder VSS
Technische Informatik:	RAM
Nach Wahl:	2 weitere aus GRIS-Katalog

Theoretische Informatik

PNL, STH, 1 aus AUK und LOS

Schwerpunkte

- Eingebettete Systeme
- Rechnernetze und Telematik

Empfohlene Ergänzungsfächer

Architektur (HfBK), Betriebswirtschaftslehre, Biochemie/Molekularbiologie, Biologie, Chemie, Geographie, Geologie, Geophysik, alle ingenieurwissenschaftlichen Studiengänge (TUHH), Mathematik, Medizin, Meteorologie, Mineralogie, Ozeanographie, Physik.

4.5 Individuelle Studienprofile

In einem individuellen Studienprofil können die einzelnen Bausteine des Hauptstudiums im Prinzip frei kombiniert werden. Der oder die Studierende trägt selbst die Verantwortung dafür, daß die für die aufbauenden Lehrveranstaltungen benötigten Grundlagen erworben werden.

Der Aufbau des Hauptstudiums muß den in 3.1 dargestellten Erfordernissen entsprechen:

- Fünf Grundlagenveranstaltungen aus dem Katalog *Grundlagen von Informatiksystemen*, davon je eine aus Angewandter, Praktischer und Technischer Informatik, die anderen wahlfrei,
- drei Grundlagenveranstaltungen aus dem Katalog *Theoretische Informatik*,

- zwei Seminare und zwei Projekte,
- ein Schwerpunkt, bestehend aus 4 SWS Schwerpunkt-Grundlagen (eine weitere Grundlagenveranstaltung, ggf. auch eine aus einem anderen Studiengang) und 12 SWS aufbauenden Veranstaltungen,
- ein Ergänzungsfach.

Die aufbauenden Veranstaltungen gehören in der Regel zu ein- und demselben Schwerpunkt eines angebotenen Studienprofils. Falls nicht, so muß bei der Beantragung sichergestellt werden, daß sich ein Prüfer bzw. eine Prüferin für den angestrebten individuellen Schwerpunkt bereit erklärt.