

Konzeption der Lehrerweiterbildung Informatik an der Freien Universität Berlin ab dem Schuljahr 2015/2016

Elfriede Fehr, Carsten Schulte und Stefan Schmidt

Stand: 15.12.2014

Grundlagen:

- Lehrkräftebildungsgesetz (LBiG) vom 7. Februar 2014
- Studien- und Prüfungsordnung des Fachbereichs Mathematik und Informatik der Freien Universität Berlin für den Bachelorstudiengang Informatik für das Lehramt vom 27. August 2014
- Kooperationsvereinbarung zwischen der Freien Universität Berlin und der Senatsverwaltung für Bildung, Jugend und Sport vom 14. August 2006

Anlagen:

1. Zertifikat-Muster
2. Kostenaufstellung
3. o. g. Kooperationsvereinbarung

Präambel

Der Fachbereich Mathematik und Informatik der Freien Universität Berlin bietet seit 1992 in Zusammenarbeit mit der Senatsverwaltung für Bildung, Jugend und Wissenschaft ein berufsbegleitendes Informatikstudium für Lehrerinnen und Lehrer an.

Der neue Lehrerweiterbildungskurs Informatik richtet sich an Lehrkräfte, die

- eine Unterrichtsbefähigung in einem weiteren Fach erwerben wollen (Erweiterung) oder
- die mit dem Erwerb der Unterrichtsbefähigung einen Wechsel des Lehramts (Ergänzung) anstreben oder
- eine Unterrichtsbefähigung in einem zweiten Fach („Quereinsteiger“) erwerben wollen.

Ziele, Inhalte und Anforderungen der neuen Weiterbildungsstudien orientieren sich an denjenigen des lehramtsbezogenen Bachelorstudiengangs und dem darauf aufbauenden ein- bzw. zweijährigen lehramtsbezogenen Masterstudiengang jeweils mit Zweitfach Informatik.

Neben dieser Orientierung der Lehrerweiterbildung Informatik am grundständigen Studiengang erhält diese aber auch ein spezifisches Profil, das an die bereits erworbenen Kompetenzen der studierenden Lehrkräfte anschließt. Die folgenden Punkte beschreiben dieses Profil.

1. Die fachwissenschaftlichen, fachdidaktischen und unmittelbar schulbezogenen Veranstaltungsziele werden in den jeweiligen Modulen explizit ausgewiesen.
2. Die Fachdidaktik enthält ein Reflexionsmodul (Analyse fachlichen Lernens), in dem die Fachinhalte der letzten drei Semester auf mögliche schulische Lernprozesse/Umsetzungen reflektiert werden sollen. Das Modul kann optional als Reflexion des eigenen Lernverlaufs oder auch als eine Art begleitende Praxisphase – im Zusammenhang mit dem Schülerlabor – umgesetzt werden. Ziel ist, die Inhalte aus Lernerperspektive wahrnehmen zu können und so fachdidaktische Kompetenzen bezüglich möglicher Lernprozesse, -hürden, -verläufe ... zu

vertiefen.

3. Die Auswahl der fachwissenschaftlichen Inhalte wurde im Vergleich zur vorherigen Fassung (vom 7. Februar 2012) nochmals anhand des Aufbaus des grundständigen Studiums gemäß der neuen Studien- und Prüfungsordnung vom 27. August 2014 geprüft und ergänzt und an die neuen Vorgaben der Senatsverwaltung für Bildung, Jugend und Wissenschaft angepasst.
4. Es gibt zwei Praktika/Projektveranstaltungen, in denen jeweils ein schulbezogenes Datenbank- bzw. Softwareprojekt erstellt und auf schulische Umsetzung bzw. Umsetzungsmöglichkeiten hin konzipiert und ausgewertet werden soll.
5. Zur Qualitätssicherung wird am Ende jedes Semesters durch ein Gremium (bestehend aus den in der Lehrerweiterbildung tätigen, abgeordneten Lehrern und den vom Fachbereich Mathematik und Informatik eingesetzten Verantwortlichen für die Lehrerweiterbildung) eine (interne) Evaluation des abgelaufenen Semesters und die Fortschreibung der Planung für das kommende Semester vorgenommen.

Organisation

Die Anmeldung von Lehrkräften der Berliner Schule erfolgt auf dem Dienstweg über die Schulleitung und die regionale Schulaufsicht des Bezirks an die Senatsverwaltung für Bildung, Jugend und Wissenschaft. Anmeldeformulare und Details der jeweiligen Ausschreibung sind im weltweiten Netz auf der Informationsseite der Senatsverwaltung zu finden.

An der berufsbegleitenden Weiterbildung teilnehmende Lehrkräfte des Landes Berlin erhalten – unabhängig von ihrem Beschäftigungsgrad – für die Dauer der Weiterbildung eine dem Studienaufwand angemessene Unterrichtsermäßigung (derzeit pro Wochentag 5-6 Unterrichtsstunden).

Die teilnehmenden Lehrkräfte immatrikulieren sich an der Freien Universität Berlin als Teilzeitstudierende und tragen die Kosten für die Immatrikulation, die das Semesterticket der BVG beinhalten.

Alle Veranstaltungen des Lehrerweiterbildungskurses finden wöchentlich an gleichen Wochentagen innerhalb der Schulhalbjahre statt. In der Regel sind das 18 Termine pro Halbjahr und Wochentag, jeweils zwischen 8 und 16 Uhr. Ausnahmen sind der jeweiligen Ausschreibung des Kurses zu entnehmen. Alle Veranstaltungen finden damit außerhalb der Schulferien, jedoch durchaus innerhalb der universitären Semesterferien statt.

Zur Bedeutung der Lehrerweiterbildung Informatik

Die Bedeutung der Informatik für Entwicklung und Wachstum der Weltwirtschaft ist unbestritten; der Stellenwert des Faches in der allgemein- und berufsbildenden Schule steht ebenfalls außer Frage.

Seit Jahrzehnten gehört in Berlin Informatik zum Spektrum der Prüfungsfächer im Abitur. Das Fach wird an über 250 Schulen in Berlin unterrichtet.

Die Anzahl der Schulen lässt erahnen, wie viele Informatiklehrer vom Land Berlin benötigt werden. Mit dem Blick auf die Tatsachen, dass in den vergangenen und kommenden Jahren ein Großteil der Berliner Lehrerinnen und Lehrer in den Ruhestand versetzt wurde bzw. wird und aus dem grundständigen Studium im Laufe der nächsten Jahre nicht mit Nachwuchs im erforderlichen Umfang zu rechnen ist, besteht weiterhin ein großer Bedarf an ausgebildeten Lehrkräften im Fach Informatik.

Die Lehrerweiterbildung bietet sich hier an, ihren Beitrag zu leisten.

Die modularisierte Struktur der Weiterbildungsstudien Informatik

In Anlehnung an die Studien- und Prüfungsordnung des Fachbereichs Mathematik und Informatik der Freien Universität Berlin für den Bachelorstudiengang Informatik für das Lehramt, bestehen die Weiterbildungsstudien Informatik aus den folgenden Modulen.

Modul	Umfang in SWS ¹	LP ¹	
Funktionale Programmierung	4 V + 4 Ü	8	
Imperative und objektorientierte Programmierung	4 V + 3 Ü	8	
Datenstrukturen und Datenabstraktion	4 V + 2 Ü	8	
Nichtsequentielle und verteilte Programmierung	4 V + 2 Ü	9	
Grundlagen der Technischen Informatik	2 V + 4 Ü/P	10	
Rechnerarchitektur, Betriebs- und Kommunikationssysteme	Rechnerarchitektur	2 V + 2 Ü	5
	Betriebssystemwerkzeuge	1 V + 1 Ü	2
	Rechnernetze	1 V + 1 S	3
Grundlagen der Theoretischen Informatik	3 V + 2 Ü	7	
Datenbanksysteme	4 V + 2 Ü	7	
Fachdidaktik Informatik	4 S	8	
Analyse fachlichen Lernens	2 S	5	
Unterrichtsbezogenes Datenbankpraktikum	1 S + 2 P	5	
Unterrichtsbezogenes Softwarepraktikum	1 S + 2 P	5	
Gesamtumfang:	64	90	

Bei erfolgreichem Abschluss der Weiterbildungsstudien im Umfang von 90 Leistungspunkten erteilt die Universität gemäß §18 des Lehrkräftebildungsgesetzes vom 07. Februar 2014 ein Zertifikat, das als Lehrbefähigung für das Fach Informatik im Sinne §18 Absatz 2 Satz 1 gilt.

Die Verteilung der Module ist abhängig von der Anzahl der Studiensemester.

¹ SWS = Semesterwochenstunde, LP = Leistungspunkte, V / Ü / P / S = Vorlesung / Übung / Praktikum / Seminar

Variante A

Der Kurs gliedert sich in zwei Kursjahre zu jeweils etwa 36 Wochen. Der zeitliche Umfang der Lehrveranstaltungen beträgt 16 Wochenstunden pro Kurshalbjahr; der Kurs findet an zwei Wochentagen von 8 bis 16 Uhr statt.

Kurshalbjahr	Module – Tag I		Module – Tag II	
	1	Funktionale Programmierung	8 SWS	Betriebssystemwerkzeuge Grundlagen der Technischen Informatik
2	Imperative und objektorientierte Programmierung Rechnerarchitektur a	7 SWS	Grundlagen der Theoretischen Informatik	5 SWS
		1 SWS	Rechnerarchitektur b	3 SWS
3	Datenstrukturen und Datenabstraktion Fachdidaktik Informatik	6 SWS	Datenbanksysteme	6 SWS
		2 SWS	Fachdidaktik Informatik	2 SWS
4	Nichtsequentielle und verteilte Programmierung Rechnernetze	6 SWS	Unterrichtsbezogenes Datenbankpraktikum	3 SWS
		2 SWS	Unterrichtsbezogenes Softwarepraktikum	3 SWS
			Analyse fachlichen Lernens	2 SWS

Variante B (Alternative)

Der Kurs gliedert sich in drei Kursjahre zu jeweils etwa 36 Wochen. Der zeitliche Umfang der Lehrveranstaltungen beträgt 10-11 Wochenstunden pro Kurshalbjahr; der Kurs findet an einem Wochentag von 8 bis 16 Uhr statt und an einem zweiten Wochentag von 16 bis 18 Uhr.

Kurshalbjahr	Module	
1	Betriebssystemwerkzeuge	– 2 SWS
	Funktionale Programmierung	– 8 SWS
	Grundlagen der Technischen Informatik a	– 2 SWS
2	Imperative und objektorientierte Programmierung	– 7 SWS
	Grundlagen der Technischen Informatik b	– 4 SWS
3	Datenstrukturen und Datenabstraktion	– 6 SWS
	Rechnerarchitektur	– 4 SWS
4	Nichtsequentielle und verteilte Programmierung	– 6 SWS
	Grundlagen der Theoretischen Informatik a	– 4 SWS
5	Grundlagen der Theoretischen Informatik b	– 1 SWS
	Datenbanksysteme	– 6 SWS
	Fachdidaktik Informatik	– 4 SWS
6	Unterrichtsbezogenes Datenbankpraktikum	– 3 SWS
	Unterrichtsbezogenes Softwarepraktikum	– 3 SWS
	Analyse fachlichen Lernens	– 2 SWS
	Rechnernetze	– 2 SWS

Allgemeine Ziele

Ziele der Lehrerweiterbildung Informatik ist die Vermittlung eines soliden, stabilen, vernetzten, fachdidaktisch fundierten und weitgehend zeitinvarianten Fachwissens im Rahmen des Fachprofils der Kultusministerkonferenz. Es soll die Lehrkräfte in die Lage versetzen,

- eigenständig Fachinhalte wissenschaftlich korrekt, altersgerecht und konform zum Rahmenlehrplan auf Schülerniveau zielgruppengerecht zu vermitteln,
- die Realisierbarkeit solcher Vorhaben realistisch einzuschätzen,
- Schülerinnen und Schüler wissenschaftspropädeutisch an das Hochschulniveau heranzuführen und ihnen einen Eindruck von Anspruchsniveau und Breite des Hochschulstudiums der Informatik zu vermitteln,
- wohldurchdachte Innovationen in den Informatikunterricht einzubringen und sich aktiv an zukünftigen Rahmenlehrplan-Erörterungen zu beteiligen.

Modulübergreifende Ziele

Die teilnehmenden Lehrkräfte sollen

- Informatik als sprachbildendes Unterrichtsfach (wie bisher z. B. Mathematik und Latein) begreifen,
- Informatik als Ingenieurwissenschaft mit den ihr inhärenten Widersprüchen zwischen Entwurfszielen und technisch notwendigen Kompromissen verstehen,
- die Bedeutung der Abstraktion kennen und die Einhaltung von Abstraktionsschranken und Invarianten beachten,
- Software auf der Basis von Spezifikationen systematisch entwickeln können,
- Einsicht in die Diskrepanz zwischen Modellierungs- und Implementierungsschicht besitzen,
- Probleme gesellschaftlicher Auswirkungen des Einsatzes globaler Informatiksysteme kennen,
- Grundsätze der informationellen Selbstbestimmung und des Datenschutzes kennen und
- wissen, wie ein sorgfältiger Umgang mit personenbezogenen, elektronisch gespeicherten Daten geleistet werden kann.

Vorbemerkungen zu fachdidaktischen und schulbezogenen Inhalten und Zielen

Die Inhalte reichen von der Einbeziehung von Unterrichtseinheiten bis zur Behandlung spezieller fachdidaktischer Fragestellungen zu Fachthemen. Das Zusammenführen eigener Erfahrungen mit einer breiten Palette von Vermittlungsformen (Diskussion, Präsentation, Exkursion, Experimentalunterricht, systematische Beobachtung, Lernen durch Lehren etc.) wird angestrebt, stehen jedoch unter dem Vorbehalt der Realisierbarkeit innerhalb des vorgegebenen engen Zeitrahmens. Das Modul „Analyse fachlichen Lernens“ wirkt dabei ergänzend. Die gesondert ausgewiesenen unterrichtsbezogenen Kombimodule „Softwarepraktikum“ und „Datenbankpraktikum“ dienen unter anderem dazu, Material für den Informatikunterricht kennen zu lernen und selbst erstellen zu können.

Die Lehrkräfte können

- Fachinhalte in ihrem Schwierigkeitsgrad beurteilen,
- begründen, ob, und wenn ja, inwieweit Fachinhalte im Schulunterricht realisierbar sind,
- Abhängigkeiten der Fachinhalte von anderen Inhalten (z.B. Voraussetzungen) angeben,
- die Bedeutsamkeit dieser Abhängigkeiten für einen logischen Stoffaufbau darlegen,
- den Stellenwert der Inhalte für den Rahmenlehrplan der Sek I / Sek II begründen,
- exemplarisch zu Fachinhalten aus Veranstaltungen der Weiterbildung Unterrichtseinheiten entwickeln,
- Literatur (Lehrbuch, Veröffentlichung, Quelle aus dem Internet) recherchieren und kritisch bewerten,
- Fachinhalte vor dem Hintergrund der allgemeinen Bildungsziele des Faches in der Bedeutsamkeit einordnen,
- bei Verfahren und Algorithmen Verständnishürden erfassen und didaktisch-methodisch Abhilfen angeben und
- Wechselbezüge einzelner Fachinhalte fachübergreifend herstellen – insbesondere zu Fachinhalten anderer Schulfächer – und ihre Wichtigkeit einschätzen.

Modulbeschreibungen

Betriebssystemwerkzeuge

Umfang: 1 V + 1 Ü (2 LP)

Voraussetzungen: keine

Inhalte:

- Zahlen und Zeichen, Bits und Bytes
- Dateisystem
- Kommandos
- Kommandointerpreter
- Reguläre Ausdrücke
- Gebrauch eines Editors

Qualifikationsziele:

Die Lehrkräfte können

- mit einer Textkonsole umgehen und sich kommandoorientiert im Dateisystem eines Rechners orientieren,
- eine Administrationsaufgabe in einer Folge von Kommandos umsetzen,
- reguläre Ausdrücke charakterisieren und z.B. zur Suche von Mustern anwenden,
- einen textorientierten Systemeditor – auch mit Hilfe regulärer Ausdrücke – effizient benutzen.

Fachdidaktik und Schulbezug:

Integrierte Behandlung einer Auswahl aus

- Dekonstruktion von Informatiksystemen
- Automatisierung von Interaktion
- Wartung von Servern

Formen aktiver Teilnahme: Schriftliche Bearbeitung der Übungsblätter und mündliche Präsentation der Lösungen in den Übungen - **keine Prüfung**

Funktionale Programmierung

Umfang: 4 V + 4 Ü (8 LP)

Voraussetzungen: keine

Inhalte:

- Syntaxdefinition (Backus-Naur)
- Mathematische Grundbegriffe (Mengensprechweise, Funktionsbegriff)
- Ausdrücke, Funktionsdefinitionen, Parameter, Musteranpassung
- Lineare und nichtlineare Rekursion, Endrekursion, Akkumulatortechnik
- Primitive Typen, Tupel, Listen, Funktionstypen
- Polymorphie
- Typsystem, Typüberprüfung und -herleitung

- Standardfunktionen, Currying, Funktionale
- Algebraische Datentypen, explizite Konstruktion von Datenstrukturen (z.B. Listen, Binärbäume, Vielwegbäume, Operatorbäume)
- Abstrakte Datentypen (Schnittstellenspezifikation, alternative Implementierungen, Benutzung der Schnittstelle, modularer Programmentwurf, Anwendungen)
- Auswertungsstrategien
- Elementare und höhere Such- und Sortieralgorithmen mit Aufwandsbetrachtungen
- primitive Rekursion, Arithmetik
- Beweis von Gesetzmäßigkeiten mit struktureller Induktion

Qualifikationsziele:

Die Lehrkräfte können im funktionalen Konzept

- elementare Algorithmen entwerfen, implementieren und testen,
- dabei rekursives Denken als Problemlösungstechnik anwenden,
- problemangemessene Datentypen definieren,
- Abstraktion durch Parametrisierung einsetzen,
- Rekursion durch geeignete Funktionale ersetzen,
- Schnittstellen spezifizieren, benutzen und implementieren,
- Algorithmen auf ihren Aufwand hin untersuchen,
- Eigenschaften von Algorithmen formal beweisen.

Fachdidaktik und Schulbezug:

Integrierte Behandlung einer Auswahl aus

- Zugänge und Blockaden bei der Anwendung von Rekursion zur Problemlösung
- Bedeutung des statistischen Typkonzepts für den Interaktionsprozess beim Programmieren
- Abstraktion durch Parametrisierung als Entwurfsmuster
- Einsatzmöglichkeiten der funktionalen Programmierung im Schulunterricht
- Die Rolle von Funktionalen und ihre Bedeutung für elegante Programmierung
- Verwendung funktionaler Notation als Pseudocode zur Problemlösung im Unterricht

Modulprüfung: Klausur

Imperative und objektorientierte Programmierung

Umfang: 4 V + 3 Ü (8 LP)

Voraussetzungen: Funktionale Programmierung

Inhalte:

Einführung in die imperative und objektorientierte Programmierung

- Grundkonzepte der imperativen Programmierung (Zustandsbegriff, Veränderungen von Variablenwerten, Kontrollstrukturen)
- Funktionen und Methoden, Parametermechanismen, Lokalität und Sichtbarkeit
- Atomare und zusammengesetzte Datentypen
- Programmtransformation: Endrekursion in Iteration

- Grundkonzepte der objektorientierten Programmierung (Objektbegriff, Spezifikation vs. Implementierung von Programmkomponenten, Modul- bzw. Klassenbegriff)
- Darstellung von Objekten im Arbeitsspeicher, Adresskonzept
- Anwendungen: Folgen, Stapel, Warteschlangen, Prioritätsschlangen (Implementierung mit Feldern und linearen Geflechten)
- Analyse von Such- und Sortieralgorithmen, O-Notation
- Berechenbarkeitsbegriff

Qualifikationsziele:

Die Lehrkräfte können

- grundlegende Prinzipien der strukturierten, imperativen Programmierung erklären und konzeptionelle Unterschiede zwischen dem deklarativen und dem imperativen Paradigma charakterisieren,
- Spezifikationen von Daten- und Programmstrukturen auf angemessenen Abstraktionsniveau erstellen und dabei die unterschiedlichen Sprachebenen bei der Formulierung von Spezifikation und Implementierung sauber abgrenzen,
- wesentliche Datentypen und Algorithmen entwerfen und analysieren – sie beherrschen dabei rekursive Techniken sicher und können eine angemessene breite Palette grundlegender Algorithmen programmiersprachenfrei formulieren und sie zur Lösung von Problemen einsetzen,
- mit einer Programmierumgebung sicher umgehen,
- exemplarisch die algorithmische Komplexität von Algorithmen angeben und begründen,
- die funktionale Programmierung für ausführbare Spezifikationen nutzen (Prototypen) und
- den von Abstraktionen geprägten funktionalen Programmierstil auch bei imperativer Programmierung pflegen.

Fachdidaktik und Schulbezug

Integrierte Behandlung einer Auswahl aus

- Grobplanung einer Unterrichtssequenz zu Sortier- oder Suchverfahren
- Laufzeitanalysen ohne Mathematik – wie geht das?
- Analyse von Unterrichtseinheiten zu den oben angegebenen Anwendungen
- Vorstellungen über den Rechneraufbau und die Auswirkungen auf das Verständnis von Wertsemantik vs. Referenzsemantik

Modulprüfung: Klausur

Datenstrukturen und Datenabstraktion

Umfang: 4 V + 2 Ü (8 LP)

Voraussetzungen: Funktionale Programmierung, Imperative und objektorientierte Programmierung

Inhalte:

Generische objektorientierte Datenstrukturen und effiziente Algorithmen

- Datenabstraktion (Geheimnisprinzip, abstrakte Datenobjekte und -typen, Spezifikation vs. Implementierung)

- Exemplarische Entwicklung wichtiger ADTs mit effizienten Repräsentationen und Operationen wie z.B. Folgen (verkettete Listen), geordnete Mengen (Halden, ausgeglichene Bäume), persistente Folgen und Mengen (sequentielle Dateien, ISAM-Dateien, B-Bäume), Operatorbäume, geometrische Objekte, Graphen (insbesondere Rückschrittverfahren); dazu jeweils Komplexitäts- und Effizienzbetrachtungen
- Klassenhierarchie, Vererbung, abstrakte Klassen, Schnittstellen
- Abstraktion durch höhere Parametrisierung (Funktionstypen, lokale Klassen, Generizität, Polymorphie)
- Entwicklung von Testumgebungen
- Technische Aspekte der Datenspeicherung

Qualifikationsziele:

Die Lehrkräfte können

- mit den Konzepten der Datenabstraktion umgehen,
- Problemstellungen der Informationsverarbeitung geeignet modellieren und daraus Spezifikationen von Daten- und Programmstrukturen auf jeweils angemessenem Abstraktionsniveau erstellen,
- wichtige abstrakte Datentypen spezifizieren und implementieren, wobei sie auf der Basis von Effizienzüberlegungen zwischen möglichen Repräsentationen entscheiden und gängige Implementierungen analysieren und ggf. aus Alternativen begründet auswählen,
- die Korrektheit der Implementierung begründen,
- Komponenten aus Bibliotheken in Programme implementieren,
- funktionale Programmierung als ausführbare Spezifikation (Prototypen) nutzen und
- kleine Softwaresysteme unter Verwendung mehrerer Komponenten entwickeln und den Aufwand dazu sachgerecht einschätzen.

Fachdidaktik und Schulbezug:

Eine ausführliche Thematisierung erfolgt im nachfolgenden unterrichtsbezogenen Softwarepraktikum.

Modulprüfung: Klausur

Grundlagen der Technischen Informatik

Umfang: 2 V + 4 Ü/P (10 LP)

Voraussetzungen: keine

Inhalte:

- Darstellung von Ziffern und Zeichen im Rechner (Ziffernkodierungen, Binärkodierungen)
- Informationsbegriff, platzeffiziente Kodierungen (z.B. Huffman), Fehlererkennung und -korrektur
- Zahldarstellungen und Arithmetik mit beschränkten Binärformaten (Ganzzahl- und Gleitkommaformate)
- Boole'sche Algebra, Schaltfunktionen, Normalformen, Minimierungsverfahren (KV, Quine-McCluskey)
- Gatter und Schaltnetze für den Rechneraufbau (u.a. Multiplexer, Addierer, ALU)

- Flipflops, Register, Speicherzellen
- A/D- und D/A-Umwandler
- Automaten (Moore- und Mealy-Automaten)
- ein eingebettetes System in Betrieb nehmen und unter Verwendung von Sensoren programmieren

Qualifikationsziele:

Die Lehrkräfte können

- fehlererkennende und -korrigierende Kodierungen entwerfen,
- ganze Zahlen und Dezimalzahlen in beschränkte maschineninterne binäre Repräsentationen umrechnen und damit verbundene Probleme aufweisen,
- Gesetze der Boole'schen Algebra zur Termvereinfachung einsetzen,
- die aus der physikalischen Realisierung der Logik entstehenden Probleme aufzeigen,
- aus einer Problemspezifikation ein Schaltnetz entwickeln, minimieren und in verschiedenen Technologien implementieren,
- eine speichernde Schaltung entwerfen und ihre Funktionsweise begründen,
- den Übergang von der analogen Welt zur digitalen Welt und umgekehrt beschreiben,
- Mealy- und Moore-Automaten mit Zustandsgraphen modellieren,
- aus einer Problemspezifikation ein einfaches Schaltwerk mit verschiedenen Flipfloptypen systematisch konstruieren und
- eine Entwicklungsumgebung für ein eingebettetes System installieren, dessen gegebene IDE nutzen und einfache Programme zwecks Interaktion mit der Umwelt entwickeln.

Fachdidaktik und Schulbezug:

Integrierte Behandlung einer Auswahl aus den Themen

- Grobplanung einer Unterrichtseinheit Schaltnetze
- Nützlichkeit von Kenntnissen der Rechnerstrukturen für den Elementarunterricht
- Rolle des Bausteinprinzips mit spezifizierten Schnittstellen in der Technischen Informatik
- Dualität zwischen Boole'scher Algebra und ihrer technischen Realisierung und deren Grenzen
- Verständnis von hardwareorientierten Informatiksystemen und ihren Wirkprinzipien
- Grobplanung einer Unterrichtseinheit zum Arduino

Modulprüfung: Klausur

Analyse fachlichen Lernens

Umfang: 2 S (5 LP)

Voraussetzungen: alle anderen vorherigen Module innerhalb des Weiterbildungskurses

Inhalt:

- Analytische Reflexion der Verwendbarkeit der gelernten Fachkonzepte in unterschiedliche Unterrichtskontexten
- Möglichkeiten der Unterrichtsgestaltung zu ausgewählten Themen der vorausgegangen drei bzw. fünf Halbjahre

- Analyse von fachlichen Lernprozessen im Informatikunterricht

Qualifikationsziele:

Die Lehrkräfte können

- Lernschwierigkeiten in einem Fachkontext analysieren und durch geeignete fachdidaktische Überlegungen beheben,
- Varianten der Unterrichtsgestaltung zu einem Fachkontext planen, vergleichen und bewerten
- fachliche Lernprozesse analysieren: Qualität des Lernens
- selbstgesteuertes Lernen initiieren (Persistenz, Momente des Erkenntnisgewinns - „Aha-Effekt“)
- kognitive Blockaden im aktuellen Lerngeschehen erkennen und ihnen mit prozessualen Einhilfen begegnen,
- Fehlvorstellungen beim Lernenden diagnostizieren und durch geeignete fachdidaktisch begründete Maßnahmen diesen entgegenwirken.

Modulprüfung: Dokumentation und Gruppenvortrag oder Präsentation

Unterrichtsbezogenes Softwarepraktikum

Umfang: 1 S + 2 P (5 LP)

Voraussetzungen: Datenstrukturen und Datenabstraktion

Inhalte:

Arbeitsteilige Erstellung eines neuen oder Weiterentwicklung eines vorhandenen Softwaresystems unter Berücksichtigung der gesellschaftlichen Implikation seines Einsatzes

- Systemanalyse und Anforderungsdefinition des ausgewählten Systems
- Objektorientierter Entwurf des gewählten Programmsystems (Klassendiagramm, Vererbung)
- Erarbeitung von Spezifikationen mit dem Schwerpunkt auf der Konstruktion mindestens zwei abstrakter Datentypen
- arbeitsteilige Implementierung von Systemkomponenten einschließlich Tests
- der Programmlebenszyklus

Zusätzlich optional eine Auswahl aus:

- Verwendung von Schnittstellen von Komponenten mit unbekannter Implementierung
- Einbeziehung nebenläufiger/verteilter Prozesse
- Einbinden von Fremdsystemen (z. B. Oberflächengestaltung, 2-/3-D-Grafik, Kommunikations- oder Datenbanksystemen)
- Systemintegration, Wiederverwendbarkeits- und Stabilitätsfragen
- Aspekte des Urheberrechts, des Datenschutzes und der informationellen Selbstbestimmung

Qualifikationsziele:

Die Lehrkräfte können

- ein didaktisch reduziertes Modell eines Programmlebenszyklus und seiner typischen Phasen skizzieren und erläutern,
- typische Probleme im Umgang mit zunehmender Formalisierung von Sprache bei Beschreibung und Konstruktion komplexer Systeme insbesondere auch mit dem Blick auf

den Schulunterricht thematisieren und berücksichtigen,

- mit softwaretechnischen Prinzipien, Methoden und Werkzeugen umgehen, sie beurteilen und einsetzen sowie deren Leistungsfähigkeit einschätzen,
- angemessen dimensionierte Komponenten eines Softwaresystems spezifizieren und implementieren und dabei den Begriff der Wiederverwendbarkeit präzisieren und sinnvoll nutzen.

Zusätzlich – je nach getroffener Auswahl – können sie auch

- Rechtskonzepte in Entwürfen für geeignete Datenstrukturen (z.B. einen ADT „SchülerIn“) modellieren,
- grundlegende Schwachstellen oder Instabilitäten komplexerer Systeme prinzipiell aufdecken,
- Grundbegriffe des Urheberrechts und des Datenschutzes aus Rechtsgrundlagen extrahieren und darstellen und den Begriff der „informationellen Selbstbestimmung“ im Kontext der Rechtsprechung erläutern.

Fachdidaktik und Schulbezug:

Integrierte Behandlung einer Auswahl aus

- Allgemeiner Projektbegriff versus Lehrprojektbegriff im Unterricht
- Dimensionierung von Lehrprojekten zur Verwendung im Schulunterricht
- Untersuchung von Projekten auf ihre Brauchbarkeit im Informatikunterricht
- Aspekte von Datenschutz und Datensicherheit in der Schule
- Wirkprinzipien größerer und komplexerer Informatiksysteme

Modulprüfung:

Abgabe der Quelltexte des erstellten Softwaresystems inklusive Dokumentation und Spezifikationen und Demonstration des Systems und seiner Komponenten im Gruppenvortrag

Grundlagen der Theoretischen Informatik

Umfang: 3 V + 2 Ü (7 LP)

Voraussetzungen: keine

Inhalte:

- Grundbegriffe (Grammatik, Sprache, Automat)
- Chomsky-Hierarchie
- Reguläre Ausdrücke, reguläre Sprachen und endliche Automaten
- Äquivalenz zwischen endlichen deterministischen und nichtdeterministischen Automaten
- Kontextfreie Grammatiken und Kellerautomaten
- Berechenbarkeitsbegriff
- Registermaschinen, Turingmaschine
- Grenzen der Berechenbarkeit (z.B. Halteproblem und andere unlösbare Probleme)

Qualifikationsziele:

Die Lehrkräfte können

- die ausgewählten Grundbegriffe der theoretischen Informatik erläutern und korrekt verwenden,

- den Bezug von formalen Sprachen und Automaten zu Konzepten der erlernten Programmiersprache herstellen,
- den Grammatik- und Ableitungsbegriff und die von einer Grammatik erzeugte Sprache definieren und Beispiele angeben,
- endliche Automaten durch Zustandsgraphen veranschaulichen und zur Beschreibung einfacher Strukturen oder Abläufe einsetzen,
- entscheiden, ob ein Wort von einem gegebenen endlichen Automaten erkannt wird,
- zu einer regulären Grammatik einen deterministischen endlichen Automaten konstruieren, der alle Wörter aus der von ihr erzeugten Sprache erkennt,
- reguläre Ausdrücke zur Beschreibung regulärer Sprachen entwickeln,
- die Äquivalenz der Ausdrucksstärke der genannten Konzepte an geeigneten einfachen Beispielen plausibel machen,
- den Begriff der (maschinellen) Berechenbarkeit präzisieren und
- die prinzipiellen Grenzen der Lösbarkeit bestimmter Probleme anhand geeigneter Beispiele aufzeigen.

Fachdidaktik und Schulbezug:

Integrierte Behandlung einer Auswahl von

- Anwendung von regulären Ausdrücken im Schulunterricht
- Automaten im Unterricht der Sek II
- Querbezug der regulären Sprachen zu Konzepten der Rechnerorganisation
- Grenzen der Anwendbarkeit regulärer Grammatiken
- Arithmetik mit Register- oder Turingmaschinen
- Grundsätzliche Bedeutung der Unlösbarkeit des Halteproblems auf der Basis einer didaktischen Reduktion

Modulprüfung: Klausur oder mündliche Prüfung

Nichtsequentielle und verteilte Programmierung

Umfang: 4 V + 2 Ü (9 LP)

Voraussetzungen: Imperative und objektorientierte Programmierung, Datenstrukturen und Datenabstraktion

Inhalte:

entsprechen denjenigen des Bachelorstudiengangs Informatik

Qualifikationsziele:

entsprechen denjenigen des Bachelorstudiengangs Informatik

Fachdidaktik und Schulbezug:

Untersuchung von Inhalten der Vorlesung auf ihre Brauchbarkeit im Informatikunterricht (didaktische Reduktion)

Modulprüfung: Klausur

Datenbanksysteme

Umfang: 4 V + 2 Ü (7 LP)

Voraussetzungen: keine

Inhalte:

- Datenunabhängigkeit, Drei-Schema-Architektur, Architektur eines DBS
- Datenbankentwurf: Phasen, ER-Modell, funktionelle Abhängigkeit, Normalformen
- Relationales Datenmodell: relationale Algebra, Relationenkalkül
- Relationale Datenhandhabungssprachen: Schemadefinition, Sichten
- Fremdschlüssel, Integritätsbedingungen
- SQL: Datendefinition, Abfragen, Anwendungsentwicklung
- Datenbankintegrität: Konsistenzsicherung, Transaktionsverwaltung, Serialisierbarkeitstheorie
- Sperrbasierte Synchronisation, Verklemmungen
- Fehlerbehandlung: Logs, Wiederanlauf
- Physische Datenorganisation
- Neuere Entwicklungen (XML, SQL3)

Qualifikationsziele:

Die Lehrkräfte können

- eine Miniwelt aus einem gegebenen Text in ein korrektes ER-Modell mit den richtigen Komplexitäten umsetzen,
- ein ER-Modell regelgerecht in ein relationales logisches Schema überführen und die erforderlichen Integritätsbedingungen formulieren,
- funktionale Abhängigkeiten aus gegebenen formal ableiten,
- einen Relationenentwurf in eine Sequenz von ausführbaren SQL-Anweisungen überführen,
- Datenbankabfragen durch korrekte Ausdrücke der Relationenalgebra formulieren und sie in den Relationenkalkülen und der Sprache Datalog ausdrücken,
- die SELECT-Anweisung zur Beantwortung auch komplexer Anfragen anwenden,
- den Nachweis für die Konfliktserialisierbarkeit konkreter Ablaufpläne führen und
- die Methoden zur Wiederherstellung des Systemzustands beschreiben und Logs analysieren.

Fachdidaktik und Schulbezug:

Eine ausführliche Thematisierung erfolgt im nachfolgenden unterrichtsbezogenen Datenbankpraktikum.

Modulprüfung: Klausur oder mündliche Prüfung

Rechnerarchitektur

Umfang: 2 V + 2 Ü (5 LP)

Voraussetzungen: keine

Inhalte:

- Architektur des von-Neumann-Rechners (Register, ALU, Steuerwerk, Datenpfade, Busse)
- Befehls- und Adressierungsarten, Befehlsausführungszyklus, Instruction Set Architecture

- Assemblerprogrammierung von Ein- und Zwei-Adress-Maschinen, Parameterübergabe bei Funktionsaufrufen per Register bzw. per Stack
- Verwaltung des Systemstacks
- Unterbrechungsbehandlung
- Pipelining, Speicherhierarchie, Cacheorganisation, virtuelle Speicherverwaltung
- Organisation des Hintergrundspeichers, Dateisystem

Qualifikationsziele:

Die Lehrkräfte können

- die wesentlichen Architekturbestandteile des von-Neumann-Rechners angeben,
- den Befehlsausführungszyklus auf Registertransferebene entwickeln,
- die Semantik von Befehlen auf Registertransferebene spezifizieren,
- einfache Algorithmen als lauffähige Assemblerprogramme implementieren (Ein- und Zwei-Adress-Maschinen),
- die Zustände des Systemstacks am Beispiel konkreter, auch rekursiver Funktionsaufrufe durchspielen,
- Varianten der Cache-Organisation skizzieren,
- die Adressumsetzung und die Behandlung von Seitenfehlern bei der virtuellen Hauptspeicherverwaltung erläutern und
- elementare Möglichkeiten der Beschleunigung von Rechnersystemen verständlich darstellen.

Fachdidaktik und Schulbezug:

Integrierte Behandlung einer Auswahl von

- Vom Nutzen der Assemblerprogrammierung für das Verständnis objektorientierter Programmierung
- Vergleich der Modellierung endlicher Automaten in Hardware und Software
- Simulationswerkzeuge zur Rechnerarchitektur und ihr Einsatz im Unterricht
- Grobplanung einer Unterrichtseinheit „Von-Neumann-Rechnermodell“ für Sek I / Sek II

Modulprüfung: Klausur

Fachdidaktik Informatik

Umfang: 4 S (8 LP)

Voraussetzungen: Algorithmen und Programmierung, Grundlagen der Technischen Informatik

Inhalte:

- Grundpositionen der Fachdidaktik, Reflexion der Vorstellungen über die Fachdisziplin Informatik
- Informatische Bildung und Allgemeinbildung
- Lernervorstellungen von Informatik: Computerwissenschaft, Computerbedienung, Expertenwissen, ...
- Soziologische Bedingungen (Genderaspekte, außerunterrichtliche Erfahrungen, Lernbiografien in der Informatik)
- Programmierkurs, die Rolle des Programmierens

- Unterrichtskonzepte, z.B. Rolle von Motivation, Interesse, Lebens- und Alltagsbezug (Kontextorientierung)
- Problemorientierung
- Dualitätsrekonstruktion: didaktische Rekonstruktion informatischen Fachwissens und dessen Verknüpfung mit allgemeinbildenden Zielen
- Konstruktivistisches Verständnis von Lernen: Lernumgebungen gestalten
- Didaktische Rekonstruktion von Fachkonzepten
- Grobplanung von Unterrichtsreihen und Feinplanung einzelner Unterrichtsstunden

Qualifikationsziele:

Die Lehrkräfte können

- grundlegende fachdidaktische Ansätze erläutern und Positionen begründet vergleichen,
- Lernerperspektiven (fachspezifisches Interesse, Lebens- und Alltagsbezug, Motivation) erläutern und in Unterrichtsplanung und -gestaltung berücksichtigen,
- verschiedene fachspezifische Unterrichtskonzepte und -techniken einsetzen,
- das Verfahren der Dualitätsrekonstruktion auf konkrete Beispiele anwenden,
- sich mit dem konstruktivistischen Verständnis von Lernen auseinandersetzen und seine Umsetzung im Informatikunterricht diskutieren,
- mit Hilfe des Rahmenlehrplans Unterrichtseinheiten grob planen, dabei den gewählten Aufbau der Einheit inhaltlich begründen und zwischen verschiedenen Planungen begründet auswählen und
- eine/zwei Unterrichtsstunde(n) zu einem vorgegebenen Thema und einer vorgegebenen Lerngruppe konkret planen und dazu einen Unterrichtsentwurf verfassen (inklusive Sachanalyse und didaktischer Reduktion).

Modulprüfung: Präsentation und Abgabe einer Dokumentation

Unterrichtsbezogenes Datenbankpraktikum

Umfang: 1 S + 2 P (5 LP)

Voraussetzungen: Datenbanksysteme

Inhalte:

Das Praktikum vertieft Fachinhalte aus dem Modul Datenbanksysteme an Beispielen, die mit der Erfahrungswelt der Schülerinnen und Schüler harmonisieren.

- Modellierung von Miniwelten
- ER-Entwurf mit Komplexitäten und Transformation in das Relationenmodell
- Anfragekonstruktion in Relationenalgebra, Tupelkalkül, Domainkalkül, SQL und Datalog
- Datendefinition, Typen, Integritätsbedingungen und Aggregatfunktionen in SQL
- Elemente der Datenbankadministration

Qualifikationsziele:

Die Lehrkräfte können

- Miniwelten im ER-Modell und im Relationenmodell korrekt modellieren,
- textuell gegebene Anfragen in SQL und andere Sprachparadigmen umsetzen und auf Korrektheit testen,

- mit den im Praktikum eingesetzten Werkzeugen souverän umgehen,
- Grundkenntnisse der Administrationskomponenten eines DBMS nachweisen.

Fachdidaktik und Schulbezug:

Integrierte Behandlung einer Auswahl von

- Unterrichtsgerechte Software und Werkzeuge für den Informatikunterricht zu Datenbanksystemen
- Erfinden und Konstruieren von Miniwelten für den Schulgebrauch
- Konstruktion von Spieldaten und Anfrageaufgaben gestuften Schwierigkeitsgrads mit Lösungen zu Miniwelten mit vorgegebenen Entwurf
- Aspekte von Datenschutz und Datensicherheit bei der Administration und Anwendung von Datenbanksystemen
- Deklarative Problemlösung mit Mengenbeschreibungen als Hilfsmittel zur Konstruktion äquivalenter SQL-Anfragen

Modulprüfung: Ausarbeitung und Präsentation

Rechnernetze

Umfang: 1 V + 1 S (3 LP)

Voraussetzungen: Rechnerarchitektur

Inhalte:

- Schichtenmodelle, Bitübertragungs- und Sicherungsschicht, Ethernet
- Netzwerkschicht, Internet-Protokolle
- Transportschicht, TCP
- Sitzungs-, Darstellungsschicht und Anwendungsschicht; Anwendungen (z.B. FTP, HTTP)
- Basiswerkzeuge zur Realisierung und Überprüfung der Netzkommunikation

Qualifikationsziele:

Die Lehrkräfte können

- Protokolle der Bitübertragungsschicht und deren Sicherung beschreiben,
- die Aufgaben der einzelnen Schichten und ihre Abhängigkeiten erläutern,
- die TCP-APIs verstehen und ihren Einsatz zur Realisierung des Kunden-Anbieter-Paradigmas beschreiben und
- eine Fehlersuche im Netz durch Anwendung elementarer Netzwerkzeuge erfolgreich durchführen.

Fachdidaktik und Schulbezug:

Integrierte Behandlung von

- rechtliche und ethische Fragen beim Einsatz von Netzanalysewerkzeugen
- Datenschutz und das Recht auf informationelle Selbstbestimmung vs. Interessen des Staates/der Strafverfolgungsbehörden oder privater Unternehmen (z.B. Facebook), Verschlüsselung, Metadaten
- Urheberrecht

Formen aktiver Teilnahme: Beteiligung an den Diskussionen und Referat – **keine Prüfung**