
Berufsbegleitende Studien im Rahmen des Quereinstiegs in das Berliner Lehramt an Integrierten Sekundarschulen, Gymnasien und Beruflichen Schulen der Fachrichtung Informatik

August 2019

Fachcurriculum

Inhaltsverzeichnis

1. Inhalte und Ziele in der Fachrichtung Informatik.....	3
2. Dauer und Gliederung der berufsbegleitenden Studien	3
3. Erwerb von Leistungspunkten und Bescheinigung von Leistungen	4
4. Wiederholung nicht erfolgreich erbrachter Leistungsnachweise.....	4
5. Modulbeschreibung*	5

Senatsverwaltung für Bildung, Jugend und Familie
Referat Lehrkräftebildung, II E

Verantwortung: Heidi Hubacek
Redaktion: Dr. Patricia Deuser

Berufsbegleitende Studien im Rahmen des Quereinstiegs in ein Berliner Lehramt in der Fachrichtung Informatik an Integrierten Sekundarschulen, Gymnasien und Beruflichen Schulen gemäß § 12 Abs. 1 Lehrkräftebildungsgesetz (LBiG) vom 7. Februar 2014

1. Inhalte und Ziele in der Fachrichtung Informatik

(1) Hauptziel der Studien ist der Erwerb der grundlegenden fachwissenschaftlichen Kenntnisse in der Fachrichtung Informatik. Die Studien enthalten zudem fachdidaktische Inhalte. Eine umfassende methodisch-didaktische Ausbildung erfolgt in dem sich an die Studien anschließenden Vorbereitungsdienst.

(2) Der Erwerb eines soliden, vernetzten und weitgehend zeitinvarianten Fachwissens soll die Teilnehmenden in die Lage versetzen, eigenständig Fachinhalte wissenschaftlich korrekt, altersgerecht und konform zum Rahmenlehrplan zielgruppengerecht zu vermitteln.

(3) Die Teilnehmenden begreifen Informatik als sprachbildendes Unterrichtsfach, können Soft- und Hardware auf der Basis von Spezifikationen systematisch entwickeln, kennen Vorteile und Risiken des Einsatzes von Informatiksystemen und diskutieren damit einhergehende gesellschaftliche Auswirkungen. Sie kennen die Grundsätze der informationellen Selbstbestimmung und des Datenschutzes.

2. Dauer und Gliederung der berufsbegleitenden Studien

(1) Die Studien dauern in der Regel vier Schulhalbjahre und orientieren sich am Ablauf des Berliner Schuljahres. Sie erfolgen berufsbegleitend.

(2) Der Kurs findet an zwei Wochentagen im Zeitfenster zwischen 8:00 Uhr und 17:30 Uhr statt. In jedem Kurshalbjahr sind Module zu belegen und mit Leistungsnachweisen erfolgreich abzuschließen.

(3) Die berufsbegleitenden Studien gliedern sich in folgende Pflichtmodule:

Schulhalbjahr	Übersicht Module	Lehrveranstaltungen	LP
1.	Betriebssystemwerkzeuge	Vorlesungen + Übungen	2
	Funktionale Programmierung	Vorlesungen + Übungen	8
	Grundlagen der Technischen Informatik	Vorlesungen + Übungen	10
2.	Rechnerarchitektur	Vorlesungen + Übungen	5
	Imperative und objektorientierte Programmierung	Vorlesungen + Übungen	8
3.	Datenstrukturen und Datenabstraktion	Vorlesungen + Übungen	8
	Datenbanksysteme	Vorlesungen + Übungen	7
4.	Unterrichtsbezogenes Softwarepraktikum	Seminar + Praktikum	5
	Unterrichtsbezogenes Datenbankpraktikum	Seminar + Praktikum	5
	Rechnernetze	Vorlesung + Seminar	3
	Gesamtzahl		61

3. Erwerb von Leistungspunkten und Bescheinigung von Leistungen

(1) Die Bedingungen für das Erreichen der Leistungspunkte sind:

- die regelmäßige Teilnahme und aktive Mitarbeit an den Lehr- und Studienveranstaltungen,
- eine intensive Vor- und Nachbereitung der Studieninhalte und
- das Erbringen der Leistungsnachweise gemäß der in dem Fachcurriculum aufgeführten Module.

(2) Die Leistungsbescheinigung weist die Bezeichnung der berufsbegleitenden Studien und der absolvierten Module mit den Leistungspunkten aus.

(3) Teilnehmende, die den letzten abzulegenden Leistungsnachweis erfolgreich erbracht haben, erhalten in der Regel am Tag der letzten Lehrveranstaltung im Schuljahr die Bescheinigung. Die Bescheinigung wird auf diesen Tag datiert.

4. Wiederholung nicht erfolgreich erbrachter Leistungsnachweise

(1) Leistungsnachweise können maximal zweimal wiederholt werden:

- Die erste Wiederholung erfolgt durch eine schriftliche Nachklausur.
- Die zweite Wiederholung kann im Rahmen einer mündlichen Fachkonsultation stattfinden.

(2) Bei nicht ausreichenden Leistungen bei der Planung, Erprobung und Auswertung eines Unterrichtsvorhabens können diese maximal zweimal in Form einer Überarbeitung der Unterrichtsentwürfe wiederholt werden.

5. Modulbeschreibung*

Modul 1 Betriebssystemwerkzeuge		
Modulart:	Pflichtmodul	
Inhalte und Qualifikationsziele des Moduls:	<p><u>Inhalte:</u></p> <ul style="list-style-type: none"> • Zahlen und Zeichen, Bits und Bytes • Dateisystem • Kommandos • Kommandointerpreter • reguläre Ausdrücke • Gebrauch eines Editors <p><u>Eine Auswahl aus fachdidaktischen und schulbezogenen Themen:</u></p> <ul style="list-style-type: none"> • Dekonstruktion von Informatiksystemen • Automatisierung von Interaktionen • Wartung von Servern <p><u>Qualifikationsziele:</u> Die Teilnehmenden können</p> <ul style="list-style-type: none"> • mit einer Textkonsole umgehen und sich kommandoorientiert im Dateisystem eines Rechners orientieren, • eine Administrationsaufgabe in eine 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. 	
Leistungsnachweis:	schriftliche Bearbeitung der Übungsblätter und mündliche Präsentation der Lösungen in den Übungen	
Veranstaltungen	Wochenstunden (à 45 Minuten)	Studienleistungen
Vorlesung	1	Regelmäßige und aktive Teilnahme
Übung	1	Regelmäßige und aktive Teilnahme

*Änderungen vorbehalten.

Modul 2 Funktionale Programmierung		
Modulart:	Pflichtmodul	
Inhalte und Qualifikationsziele des Moduls:	<p><u>Inhalte:</u></p> <ul style="list-style-type: none"> • 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 <p><u>Eine Auswahl aus fachdidaktischen und schulbezogenen Themen:</u></p> <ul style="list-style-type: none"> • 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 <p><u>Qualifikationsziele:</u> Die Teilnehmenden können im funktionalen Kontext</p> <ul style="list-style-type: none"> • elementare Algorithmen entwerfen, implementieren und testen, • dabei rekursives Denken als Problemlösungstechnik anwenden, • problemangemessene Datentypen definieren, • Abstraktion durch Parametrisierung einsetzen, • Rekursion ggf. durch geeignete Funktionale ersetzen, • Schnittstellen spezifizieren, benutzen und implementieren, • Algorithmen auf ihren Aufwand hin untersuchen, • Eigenschaften von Algorithmen formal beweisen. 	
	Leistungsnachweis	Klausur
Veranstaltungen	Wochenstunden (à 45 Minuten)	Studienleistung
Vorlesung	4	Regelmäßige und aktive Teilnahme

Übung	4	Regelmäßige und aktive Teilnahme
-------	---	----------------------------------

Modul 3 Grundlagen der Technischen Informatik	
Modulart:	Pflichtmodul
Inhalte und Qualifikationsziele des Moduls:	<p><u>Inhalte:</u></p> <ul style="list-style-type: none"> • Darstellungen 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) • Boolesche 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-Wandler • Automaten (Moore- und Mealy-Automaten) • ein eingebettetes System in Betrieb nehmen und unter Verwendung von Sensoren programmieren <p><u>Eine Auswahl aus fachdidaktischen und schulbezogenen Themen:</u></p> <ul style="list-style-type: none"> • Grobplanung einer Unterrichtseinheit Schaltnetze • Nützlichkeit von Kenntnissen der Rechnerstrukturen für den Elementarunterricht • Rolle des Bausteinprinzips mit spezifizierten Schnittstellen in der Technischen Informatik • Dualität zwischen Boolescher Algebra und ihrer technischen Realisierung und deren Grenzen • Verständnis von hardwareorientierten Informatiksystemen und ihren Wirkprinzipien • Grobplanung einer Unterrichtseinheit zum Arduino <p><u>Qualifikationsziele:</u> Die Teilnehmenden können</p> <ul style="list-style-type: none"> • fehlererkennende und -korrigierende Kodierungen entwerfen, • ganze Zahlen und Dezimalzahlen in beschränkte maschineninterne binäre Repräsentationen umrechnen und die damit verbundenen Probleme aufzeigen, • 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

	<ul style="list-style-type: none"> eine Entwicklungsumgebung für ein eingebettetes System installieren, dessen gegebene IDE nutzen und einfache Programme zwecks Interaktion mit der Umwelt entwickeln. 	
Leistungsnachweis:	Klausur	
Veranstaltungen	Wochenstunden (à 45 Minuten)	Studienleistungen
Vorlesung	2	Regelmäßige und aktive Teilnahme
Übung/Praktikum	4	Regelmäßige und aktive Teilnahme

Modul 4 Imperative und objektorientierte Programmierung	
Modulart:	Pflichtmodul
Voraussetzung	Funktionale Programmierung
Inhalte und Qualifikationsziele des Moduls:	<p><u>Inhalte:</u> Einführung in die imperative und objektorientierte Programmierung</p> <ul style="list-style-type: none"> • 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 <p><u>Eine Auswahl aus fachdidaktischen und schulbezogenen Themen:</u></p> <ul style="list-style-type: none"> • Grobplanung einer Unterrichtssequenz zu Sortier- oder Suchverfahren • Laufzeitanalysen ohne Mathematik - wie geht das? • Analyse von Unterrichtseinheiten zu den o.a. Anwendungen • Vorstellungen über den Rechneraufbau und die Auswirkungen auf das Verständnis von Wertsemantik vs. Referenzsemantik <p><u>Qualifikationsziele:</u> Die Teilnehmenden können</p> <ul style="list-style-type: none"> • 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.	
Leistungsnachweis:	Klausur	
Veranstaltungen	Wochenstunden (à 45 Minuten)	Studienleistungen
Vorlesung	4	Regelmäßige und aktive Teilnahme
Übung	3	Regelmäßige und aktive Teilnahme

Modul 5 Rechnerarchitektur		
Modulart:	Pflichtmodul	
Inhalte und Qualifikationsziele des Moduls:	Inhalte: <ul style="list-style-type: none"> • 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 	
	Eine Auswahl aus fachdidaktischen und schulbezogenen Themen: <ul style="list-style-type: none"> • vom Nutzen der Assemblerprogrammierung für das Verständnis objektorientierter Programmierung • Vergleich der Modellierung endlicher Automaten in Hardware und Software • Simulationswerkzeuge zur Rechnerorganisation und ihr Einsatz im Unterricht • Grobplanung einer Unterrichteinheit "Von Neumann-Rechnermodell" für Sekundarstufe I und II 	
	Qualifikationsziele: Die Teilnehmenden können <ul style="list-style-type: none"> • 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. 	
Leistungsnachweis:	Klausur	
Veranstaltungen	Wochenstunden (à 45 Minuten)	Studienleistungen
Vorlesung	2	Regelmäßige und aktive Teilnahme
Übung	2	Regelmäßige und aktive Teilnahme

Modul 6 Datenstrukturen und Datenabstraktion		
Modulart:	Pflichtmodul	
Voraussetzung	Funktionale Programmierung, Imperative und objektorientierte Programmierung	
Inhalte und Qualifikationsziele des Moduls:	<p><u>Inhalte:</u> Generische objektorientierte Datenstrukturen und effiziente Algorithmen</p> <ul style="list-style-type: none"> • 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- bzw. Effizienzüberlegungen • Klassenhierarchie, Vererbung, abstrakte Klassen, Schnittstellen • Abstraktion durch höhere Parametrisierung (Prozedurtypen, lokale Klassen, Generizität, Polymorphie) • Entwicklung von Testumgebungen • technische Aspekte der Datenspeicherung <p><u>Fachdidaktische und schulbezogene Themen:</u> Eine ausführliche Thematisierung erfolgt im nachfolgenden unterrichtsbezogenen Softwarepraktikum.</p> <p><u>Qualifikationsziele:</u> Die Teilnehmenden können</p> <ul style="list-style-type: none"> • 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 von Implementierungen begründen, • Komponenten aus Bibliotheken in Programme integrieren, • funktionale Programmierung als ausführbare Spezifikationen (Prototypen) nutzen und • kleine Softwaresysteme unter Verwendung mehrerer Komponenten entwickeln und den Aufwand dazu sachgerecht einschätzen. 	
Leistungsnachweis:	Klausur	
Veranstaltungen	Wochenstunden (à 45 Minuten)	Studienleistungen
Vorlesung	4	Regelmäßige und aktive Teilnahme
Übung	2	Regelmäßige und aktive Teilnahme

Modul 7 Datenbanksysteme		
Modulart:	Pflichtmodul	
Inhalte und Qualifikationsziele des Moduls:	<p><u>Inhalte:</u></p> <ul style="list-style-type: none"> • Datenunabhängigkeit, Drei-Schema-Architektur, Architektur eines DBS • Datenbankentwurf: Phasen, ER-Modell, funktionale 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) <p><u>Fachdidaktische und schulbezogene Themen:</u> Eine ausführliche Thematisierung erfolgt im nachfolgenden unterrichtsbezogenen Datenbankpraktikum.</p> <p><u>Qualifikationsziele:</u> Die Teilnehmenden können</p> <ul style="list-style-type: none"> • eine Miniwelt aus einem gegebenen Text in ein korrektes ER-Modell mit den richtigen Komplexitäten umsetzen, • ein ER-Modell regelgerecht in einen 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, • Datenbankanfragen als korrekte Ausdrücke der Relationenalgebra formulieren und sie in den Relationenkalkülen und der Sprache Datalog ausdrücken, • die SELECT-Anweisung zur Beantwortung auch komplexerer Anfragen anwenden, • den Nachweis für die Konfliktserialisierbarkeit konkreter Ablaufpläne führen und • die Methoden zur Wiederherstellung des Systemzustands beschreiben und Logs analysieren. 	
	Leistungsnachweis:	Klausur oder mündliche Prüfung
Veranstaltungen	Wochenstunden (à 45 Minuten)	Studienleistungen
Vorlesung	4	Regelmäßige und aktive Teilnahme
Übung	2	Regelmäßige und aktive Teilnahme

Modul 8 Unterrichtsbezogenes Softwarepraktikum	
Modulart:	Pflichtmodul
Inhalte und Qualifikationsziele des Moduls:	<p><u>Inhalte:</u></p> <ul style="list-style-type: none"> • arbeitsteilige Erstellung eines neuen oder Weiterentwicklung eines vorhandenen Softwaresystems unter Berücksichtigung der gesellschaftlichen Implikationen 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 Test • der Programmlebenszyklus <p>Eine Auswahl aus folgenden Themen:</p> <ul style="list-style-type: none"> • Verwendung von Schnittstellen von Komponenten mit unbekannter Implementierung • Einbeziehung nebenläufiger/verteilter Prozesse • Einbindung von Fremdsystemen (z.B. Oberflächengestaltung, 2-/3-D-Graphik, Kommunikations- oder Datenbanksysteme), • Systemintegration, Wiederverwendbarkeits- und Stabilitätsfragen • Aspekte des Urheberrechts, des Datenschutzes und der informationellen Selbstbestimmung. <p><u>Fachdidaktische und schulbezogene Themen:</u></p> <ul style="list-style-type: none"> • allgemeiner Projektbegriff versus Lehrprojektbegriff im Unterricht • Dimensionierung von Lehrprojekten zur Verwendung im Unterricht • Untersuchung von Projekten auf ihre Brauchbarkeit im Informatikunterricht • Aspekte von Datenschutz und Datensicherheit in der Schule Wirkprinzipien größerer und komplexerer Informatiksysteme <p><u>Qualifikationsziele:</u> Die Teilnehmenden können</p> <ul style="list-style-type: none"> • 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 softwaretechnische 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. <p>Zusätzlich - je nach getroffener Auswahl - können sie auch</p>

	<ul style="list-style-type: none"> • 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. 	
Leistungsnachweis:	Abgabe der Quelltexte des erstellten Softwaresystems inklusive Dokumentation und Spezifikationen und Demonstration des Systems und seiner Komponenten im Gruppenvortrag	
Veranstaltungen	Wochenstunden (à 45 Minuten)	Studienleistungen
Vorlesung	1	Regelmäßige und aktive Teilnahme
Praktikum	2	Regelmäßige und aktive Teilnahme

Modul 9 Unterrichtsbezogenes Datenbankpraktikum		
Modulart:	Pflichtmodul	
Voraussetzung:	Datenbanksysteme	
Inhalte und Qualifikationsziele des Moduls:	<p><u>Inhalte:</u> Das Praktikum vertieft Fachinhalte aus dem Modul Datenbanksysteme an Beispielen, die mit der Erfahrungswelt der Schülerinnen und Schüler harmonisieren.</p> <ul style="list-style-type: none"> • 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 <p><u>Fachdidaktische und schulbezogene Themen:</u></p> <ul style="list-style-type: none"> • 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 <p><u>Qualifikationsziele:</u> Die Teilnehmenden können</p> <ul style="list-style-type: none"> • 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. 	
Leistungsnachweis:	Ausarbeitung und Präsentation	
Veranstaltungen	Wochenstunden (à 45 Minuten)	Studienleistungen
Seminar	1	Regelmäßige und aktive Teilnahme
Praktikum	2	Regelmäßige und aktive Teilnahme

Modul 10 Rechnernetze		
Modulart:	Pflichtmodul	
Voraussetzung:	Rechnerarchitektur	
Inhalte und Qualifikationsziele des Moduls:	<p><u>Inhalte:</u></p> <ul style="list-style-type: none"> • Schichtenmodelle, Bitübertragungs- und Sicherungsschicht, Ethernet • Netzwerkschicht, Internet-Protokolle • Transportschicht, TCP • Sitzungs-, Darstellungs- und Anwendungsschicht; Anwendungen (z.B. FTP, HTTP) • Basiswerkzeuge zur Realisierung und Überprüfung der Netzkommunikation <p><u>Fachdidaktische und schulbezogene Themen:</u></p> <p>Integrierte Behandlung von</p> <ul style="list-style-type: none"> • rechtlichen und ethischen Fragen beim Einsatz von Netzanalysewerkzeugen • Datenschutz und Recht auf informationelle Selbstbestimmung vs. Interessen des Staates/der Strafverfolgungsbehörden oder privater Unternehmen (z.B. Facebook), Verschlüsselung, Metadaten • Urheberrecht <p><u>Qualifikationsziele:</u> Die Teilnehmenden können</p> <ul style="list-style-type: none"> • 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. 	
Leistungsnachweis:	Referat und Beteiligung an den Diskussionen	
Veranstaltungen	Wochenstunden (à 45 Minuten)	Studienleistungen
Vorlesung	1	Regelmäßige und aktive Teilnahme
Seminar	1	Regelmäßige und aktive Teilnahme