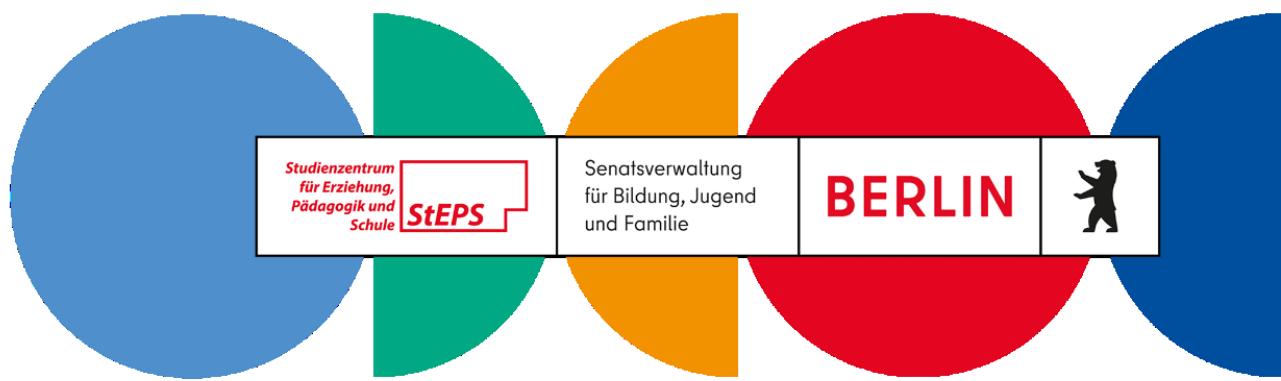


FACHCURRICULUM

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



Studienzentrum
für Erziehung,
Pädagogik und
Schule

StEPS

Senatsverwaltung
für Bildung, Jugend
und Familie

BERLIN



Impressum

Herausgeberin

Studienzentrum für Erziehung, Pädagogik und Schule
(StEPS) der Senatsverwaltung für Bildung, Jugend und Familie
Fachgruppe für die Berufsbegleitende Weiterbildung
II E 4
Georgenstraße 35
10117 Berlin

www.berlin.de/sen/bjf

Redaktion

SenBJF, Fachgruppe II E 4

Gestaltung

SenBJF, Fachgruppe II E

Auflage

Juni 2023, Rev03

Inhalt

1. 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
1.1. Inhalte und Ziele in der Fachrichtung Informatik	1
1.2. Dauer und Gliederung der berufsbegleitenden Studien	2
1.3. Nachweis und Bescheinigung von Leistungen	3
1.4. Wiederholung nicht erfolgreich erbrachter Leistungsnachweise	3
1.5. Modulbeschreibung	4

1. 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

1.1. 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.

1.2. 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 von 8:00 Uhr bis 17:30 Uhr statt.
3. Die berufsbegleitenden Studien gliedern sich in folgende Pflichtmodule:

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

1.3. Nachweis und Bescheinigung von Leistungen

1. Die berufsbegleitenden Studien gelten als erfolgreich absolviert, wenn:
 - an den Lehr- und Lernveranstaltungen regelmäßig teilgenommen und an den gestellten Anforderungen aktiv mitgewirkt wurde,
 - die Studieninhalte in Selbststudienzeiten vor- und nachbereitet wurde,
 - die studienbegleitenden Leistungsnachweise und Prüfungsleistungen für die Module erbracht wurden.
2. Bei Abwesenheit müssen versäumte Inhalte nachgearbeitet und nach Rücksprache mit den Lehrbeauftragten mündlich oder schriftlich nachgewiesen werden.
3. Die Bescheinigung weist den Erfolg der berufsbegleitenden Studien und der entsprechend absolvierten Module aus.
4. 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.
5. Leistungen werden als
 - A) studienbegleitende Leistungsnachweise erbracht. Diese belegen die im Verlauf der Lehr- und Lernveranstaltungen der einzelnen Module erworbenen Kenntnisse und Fähigkeiten sowie den Lernzuwachs bezogen auf die vermittelten fachwissenschaftlichen und fachdidaktischen Inhalte. Sie werden z.B. in Form von Übungen, Anleitungen, Entwicklungen, Trainings, Lernchancen, Feedbackschleifen, Hausarbeiten, Seminararbeiten, Referate, Präsentationen, Projekte, Portfolios, Lerntagebücher oder auch Protokolle durchgeführt.
 - B) abschließende Prüfungsleistungen erbracht. Diese belegen, dass die in den Fachcurricula beschriebenen Qualifikations- und Kompetenzziele erreicht wurden, insbesondere die in den Modulen vermittelten Inhalte und Methoden in den wesentlichen Zusammenhängen beherrschen und die erworbenen Kenntnisse und Fähigkeiten angewendet werden können. Sie werden z.B. in Form von Klausuren, mündlichen Prüfungen, Tests, Abschlussarbeiten, Essays, Kolloquien (Fachkonsultationen), Einzel- oder Gruppenprüfungen oder auch Podiumsdiskussionen) durchgeführt.
6. Über die Form der Leistungsnachweise wird zu Beginn der Studien oder ggf. spätestens zu Beginn des jeweiligen Moduls durch die Lehrbeauftragten informiert.

1.4. Wiederholung nicht erfolgreich erbrachter Leistungsnachweise

1. Studienbegleitende Leistungsnachweise und abschließende Prüfungsleistungen können maximal zweimal wiederholt werden.
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.

1.5. Modulbeschreibung*

Modul 1: Betriebssystemwerkzeuge		
Modulart:	Pflichtmodul	
Inhalte:	<ul style="list-style-type: none"> • Zahlen und Zeichen, Bits und Bytes • Dateisystem • Kommandos • Kommandointerpreter • reguläre Ausdrücke • Gebrauch eines Editors <p>Eine Auswahl aus fachdidaktischen und schulbezogenen Themen:</p> <ul style="list-style-type: none"> • Dekonstruktion von Informatiksystemen • Automatisierung von Interaktionen • Wartung von Servern 	
Qualifikationsziele des Moduls:	<p>Die Teilnehmenden</p> <ul style="list-style-type: none"> • können 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. 	
Studienbegleitende Leistungsnachweise/ Prüfungsleistungen:	<ul style="list-style-type: none"> • schriftliche Bearbeitung der Übungsblätter • Präsentation der Lösungen in den Übungen 	
Veranstaltungen	Wochenstunden (à 45 Minuten)	Studienleistung
Vorlesung	1	Regelmäßige und aktive Teilnahme
Übung	1	Regelmäßige und aktive Teilnahme

* Änderungen vorbehalten

Modul 2: Funktionale Programmierung

Modulart:

Pflichtmodul

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

Eine Auswahl aus fachdidaktischen und schulbezogenen Themen:

- 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

Qualifikationsziele des Moduls:	<p>Die Teilnehmenden</p> <ul style="list-style-type: none"> • können im funktionalen Kontext 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.
Studienbegleitende Leistungsnachweise/ Prüfungsleistungen:	<ul style="list-style-type: none"> • Regelmäßige und aktive Teilnahme • 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:	<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>Eine Auswahl aus fachdidaktischen und schulbezogenen Themen:</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

Qualifikationsziele des Moduls:	<p>Die Teilnehmenden</p> <ul style="list-style-type: none"> • können 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 • eine Entwicklungsumgebung für ein eingebettetes System installieren, dessen gegebene IDE nutzen und einfache Programme zwecks Interaktion mit der Umwelt entwickeln.
Studienbegleitende Leistungsnachweise/ Prüfungsleistungen:	<ul style="list-style-type: none"> • Regelmäßige und aktive Teilnahme • Klausur

Veranstaltungen	Wochenstunden (à 45 Minuten)	Studienleistung
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:	<p>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>Eine Auswahl aus fachdidaktischen und schulbezogenen Themen:</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

Qualifikationsziele des Moduls:	<p>Die Teilnehmenden</p> <ul style="list-style-type: none"> • 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.
Studienbegleitende Leistungsnachweise/ Prüfungsleistungen:	<ul style="list-style-type: none"> • Regelmäßige und aktive Teilnahme • Klausur

Veranstaltungen	Wochenstunden (à 45 Minuten)	Studienleistung
Vorlesung	4	Regelmäßige und aktive Teilnahme
Übung	3	Regelmäßige und aktive Teilnahme

Modul 5: Rechnerarchitektur

Modulart:	Pflichtmodul	
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 <p>Eine Auswahl aus fachdidaktischen und schulbezogenen Themen:</p> <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 Unterrichtseinheit "Von Neumann-Rechnermodell" für Sekundarstufe I und II 	
Qualifikationsziele des Moduls:	<p>Die Teilnehmenden</p> <ul style="list-style-type: none"> • 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. 	
Studienbegleitende Leistungsnachweise/ Prüfungsleistungen:	<ul style="list-style-type: none"> • Regelmäßige und aktive Teilnahme • Klausur 	
Veranstaltungen	Wochenstunden (à 45 Minuten)	Studienleistung
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:	<p>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. Effizienzbetrachtungen • Klassenhierarchie, Vererbung, abstrakte Klassen, Schnittstellen • Abstraktion durch höhere Parametrisierung (Prozedurtypen, lokale Klassen, Generizität, Polymorphie) • Entwicklung von Testumgebungen • technische Aspekte der Datenspeicherung <p>Fachdidaktische und schulbezogene Themen:</p> <ul style="list-style-type: none"> • Eine ausführliche Thematisierung erfolgt im nachfolgenden unterrichtsbezogenen Softwarepraktikum.
Qualifikationsziele des Moduls:	<p>Die Teilnehmenden</p> <ul style="list-style-type: none"> • 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 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.
Studienbegleitende Leistungsnachweise/ Prüfungsleistungen:	<ul style="list-style-type: none"> • Regelmäßige und aktive Teilnahme • Klausur

Veranstaltungen	Wochenstunden (à 45 Minuten)	Studienleistung
Vorlesung	4	Regelmäßige und aktive Teilnahme
Übung	2	Regelmäßige und aktive Teilnahme

Modul 7: Datenbanksysteme

Modulart:	Pflichtmodul
Inhalte:	<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>Fachdidaktische und schulbezogene Themen:</p> <ul style="list-style-type: none"> • Eine ausführliche Thematisierung erfolgt im nachfolgenden unterrichtsbezogenen Datenbankpraktikum.
Qualifikationsziele des Moduls:	<p>Die Teilnehmenden</p> <ul style="list-style-type: none"> • 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 Relationentwurf in eine Sequenz von ausführbaren SQL-Anweisungen überführen, • Datenbankabfragen 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.
Studienbegleitende Leistungsnachweise/ Prüfungsleistungen:	<ul style="list-style-type: none"> • Regelmäßige und aktive Teilnahme • Klausur oder mündliche Prüfung

Veranstaltungen	Wochenstunden (à 45 Minuten)	Studienleistung
Vorlesung	4	Regelmäßige und aktive Teilnahme
Übung	2	Regelmäßige und aktive Teilnahme

Modul 8: Unterrichtsbezogenes Softwarepraktikum

Modulart:	Pflichtmodul
Inhalte:	<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>Auswahl folgender 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>Fachdidaktische und schulbezogene Themen:</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>Qualifikationsziele des Moduls:</p>	<p>Die Teilnehmenden</p> <ul style="list-style-type: none"> • 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 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.
<p>Studienbegleitende Leistungsnachweise/ Prüfungsleistungen:</p>	<ul style="list-style-type: none"> • Regelmäßige und aktive Teilnahme • Abgabe der Quelltexte des erstellten Softwaresystems inklusive Dokumentation und Spezifikationen und Demonstration des Systems und seiner Komponenten im Gruppenvortrag

Veranstaltungen	Wochenstunden (à 45 Minuten)	Studienleistung
Vorlesung	1	Regelmäßige und aktive Teilnahme
Praktikum	2	Regelmäßige und aktive Teilnahme

Modul 9: Unterrichtsbezogenes Datenbankpraktikum

Modulart:	Pflichtmodul
Voraussetzung:	Datenbanksysteme
Inhalte:	<p>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, Domäinkalkül, SQL und Datalog • Datendefinition, Typen, Integritätsbedingungen und Aggregatfunktionen in SQL • Elemente der Datenbankadministration <p>Fachdidaktische und schulbezogene Themen:</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
Qualifikationsziele des Moduls:	<p>Die Teilnehmenden</p> <ul style="list-style-type: none"> • 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.
Studienbegleitende Leistungsnachweise/ Prüfungsleistungen:	<ul style="list-style-type: none"> • Regelmäßige und aktive Teilnahme • Ausarbeitung und Präsentation

Veranstaltungen	Wochenstunden (à 45 Minuten)	Studienleistung
Seminar	1	Regelmäßige und aktive Teilnahme
Praktikum	2	Regelmäßige und aktive Teilnahme

Modul 10: Rechnernetze

Modulart:	Pflichtmodul	
Voraussetzung:	Rechnerarchitektur	
Inhalte:	<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>Fachdidaktische und schulbezogene Themen:</p> <ul style="list-style-type: none"> • Integrierte Behandlung von • 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 	
Qualifikationsziele des Moduls:	<p>Die Teilnehmenden</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. 	
Studienbegleitende Leistungsnachweise/ Prüfungsleistungen:	<ul style="list-style-type: none"> • Regelmäßige und aktive Teilnahme • Referat und Beteiligung an den Diskussionen 	
Veranstaltungen	Wochenstunden (à 45 Minuten)	Studienleistung
Vorlesung	1	Regelmäßige und aktive Teilnahme
Seminar	1	Regelmäßige und aktive Teilnahme

www.berlin.de/sen/bjf

Senatsverwaltung
für Bildung, Jugend
und Familie

BERLIN

