



SCHULINTERNES CURRICULUM FÜR DAS FACH
INFORMATIK
KARDINAL-VON-GALEN-GYMNASIUM MÜNSTER

Stand: September 2018

1. Einleitung

Das Curriculum orientiert sich am Kernlernplan für das Fach Informatik¹. Es wurde im Arbeitskreis Informatik der Schulen des Bistums Münster in seinen wesentlichen Bestandteilen erarbeitet und skizziert die einzelnen Unterrichtsvorhaben für die Sekundarstufe II.

Die im Kernlehrplan verankerten Kompetenzen werden in den einzelnen Inhaltsfeldern aufgegriffen. Dabei werden jeweils die folgenden Abkürzungen verwendet:

- Argumentieren (A)
- Modellieren (M)
- Darstellen (D) und Implementieren (I)
- Kommunizieren und Kooperieren (K).

Dieses Curriculum ist Grundlage für den Oberstufenunterricht ab dem Abiturjahrgang 2017.

2. Organisation

Am Kardinal-von-Galen-Gymnasium wird Informatik in der Sekundarstufe II als neu einsetzendes Fach unterrichtet.

3. Leistungsbewertung

In der Einführungsphase und der Qualifikationsphase werden pro Halbjahr zwei 90-minütige Klausuren geschrieben. Die schriftlichen Ergebnisse gehen zu 50% in die Gesamtnote ein.

Die Anforderungen an die sonstige Mitarbeit orientieren sich im Wesentlichen an dem Kompetenzraster auf der folgenden Seite.

4. Inhalte

Ab Seite 2 werden die einzelnen Unterrichtssequenzen in folgender Reihenfolge aufgeführt.

- Kompetenzerwartung für die *Sonstige Mitarbeit* im Informatikunterricht
- Lehrplan der Einführungsphase
- Lehrplan für den Grundkurs der Qualifikationsphase

Optionale Inhalte sind mit * markiert.

Zu den Themen werden jeweils auch unterrichtliche Beispiele als Vorschläge aufgeführt, die zum Teil durch Materialien auf der Schulbistumsplattform unter dem Arbeitskreis Informatik oder durch Links weiter ausgeführt werden. Dieser Materialpool soll den Austausch unter den Fachkollegen der angeschlossenen Schulen weiter fördern. Einige der aufgeführten Unterrichtsbeispiele findet man auch in den Ergänzungsmaterialien auf QuaLis:<http://www.schulentwicklung.nrw.de/lehrplaene/lehrplannavigator-s-ii/gymnasiale-oberstufe/informatik/hinweise-und-beispiele/hinweise-und-beispiele.html>

¹ Kernlernplan für die Sekundarstufe II Gymnasium / Gesamtschule in NRW Informatik, 2013

Kompetenzerwartungen für die Sonstige Mitarbeit im Informatikunterricht

	Stufe 1	Stufe 2	Stufe 3	Stufe 4
Grundhaltung				
Pünktlichkeit	Kommt häufig zu spät.	Immer pünktlich.	Kommt immer pünktlich und hat ihre Unterlagen auf dem Tisch.	Kommt immer pünktlich und ist arbeitsbereit.
Beteiligung am Unterrichtsgespräch	Nimmt nie unaufgefordert am Unterrichtsgespräch teil.	Nimmt selten am Unterrichtsgespräch teil.	Nimmt regelmäßig am Unterrichtsgespräch teil.	Nimmt regelmäßig am Unterrichtsgespräch teil; hat gute Ideen u. zeigt Eigeninitiative.
Interesse / Neugier	Zeigt wenig Interesse an neuen Themen, abgelenkt durch fachfremde Aktivitäten.	Zeigt gelegentlich Interesse an neuen Themen.	Zeigt generell Interesse an neuen Themen; hat ausgeprägte Schwerpunkte.	Hat eine aufgeschlossene, neugierige Lernhaltung; bringt eigene Ideen und Themenschwerpunkte ein.
Selbstständigkeit				
Selbstständiges Arbeiten	Hält andere von der Arbeit ab; redet mit anderen statt zu arbeiten; muss häufiger ermahnt werden; Aufträge werden nachlässig erledigt.	Arbeitet meist nur auf Aufforderung, stört andere nicht bei der Arbeit.	Arbeitet auf Aufforderung regelmäßig u. ausdauernd; zuverlässige Erledigung der Arbeitsaufträge; fragt, wenn es notwendig ist.	Bleibt ohne Ermahnung ausdauernd bei der Arbeit; arbeitet ruhig und nutzt dabei selbstständig Notizen und Arbeitsunterlagen.
Eigeninitiative	Hat Schwierigkeiten mit der Arbeit zu beginnen; fragt nicht um Hilfe; holt Rückstand nach Abwesenheit nicht selbstständig auf; nimmt nicht teil; nur sporadische Beiträge.	Nimmt selten teil; arbeitet nur nach Aufforderung, beginnt verzögert mit der Bearbeitung gestellter Arbeitsaufträge; fragt nur selten nach Hilfe.	Nimmt rege teil, wenn sie dazu aufgefordert wird; beginnt umgehend mit gestellten Arbeitsaufträgen; arbeitet die meiste Zeit ernsthaft.	Antwortet stets eigeninitiativ auf Fragen; fragt sinnvoll nach, beginnt umgehend mit allen gestellten Arbeitsaufträgen; hilft anderen; weiß, was zu tun ist und tut es.
Hausaufgaben	Hausaufgaben meistens unvollständig.	Hausaufgaben meistens vollständig.	Hausaufgaben normalerweise vollständig und korrekt.	Hausaufgaben immer vollständig, z.T. weitere Arbeiten zu Hause erledigt.
Arbeitsorganisation	Arbeitsmaterialien oft nicht vollständig dabei und/oder in ungeordnetem Zustand.	Arbeitsmaterial normalerweise vorhanden, u.U. in ungeordnetem Zustand.	Arbeitsunterlagen in der Regel vorhanden u. meist in ordentlichem Zustand.	Arbeitsmaterialien stets vorhanden und in ordentlichem Zustand.
Methodenkompetenz				
Gruppen- und Partnerarbeit	Hält andere oft von der Arbeit ab; schwierige Partnerin in GA.	Bringt sich nur wenig ein; stört andere aber nicht; nur selten produktive Beiträge.	Arbeitet kooperativ und folgt bereitwillig anderen.	Arbeitet kooperativ, respektvoll; übernimmt Führungsrolle in der Gruppe.
Recherche	Weiß nicht, wo sie sich Informationen beschaffen kann, braucht umfangreiche Hilfe.	Kann mit Hilfestellung Informationen beschaffen.	Weiß meistens, wo sie Hilfe beschaffen kann.	Kann schnell und zielgerichtet Informationen beschaffen und auswerten.
Fachspezifische Aspekte				
<u>Rechnerarbeit:</u> Umsetzung von Algorithmen, Verfügbarkeit der erlernten Werkzeuge*	Hat Schwierigkeiten bei der Umsetzung der Basis-Werkzeuge* in der erlernten Programmiersprache; kann besprochene Algorithmen nur mit kleinschrittiger Hilfestellung in Kontrollstrukturen* u. dann auf die Programmierumgebung übertragen.	Kann die grundlegenden Werkzeuge* in der erlernten Programmierumgebung mit konkreter Hilfestellung korrekt anwenden; kann besprochene Algorithmen mit Hilfestellung in Kontrollstrukturen* und dann in die Programmierumgebung übertragen.	Kann die besprochenen Algorithmen selbstständig am Rechner umsetzen. Erlernte Werkzeuge* sind verfügbar oder können eigenständig wieder aktiviert werden.	Sehr selbstständige, effektive Umsetzung der besprochenen sowie auch darüber hinausgehender Algorithmen am Rechner; Sicherer Umgang mit der Programmierumgebung, erlernte Werkzeuge sind hierbei stets verfügbar.
<u>Rechnerarbeit:</u> Durchhaltevermögen, Kreativität, Fehlersuche	Gibt schnell auf, kann die Fehlerkommentare des Compilers nicht zur selbstständigen Lösung nutzen.	Kann syntaktische Fehler mit Hilfe der Fehlerkommentare des Compilers eigenständig beheben; benötigt noch häufig Hilfe bei der Behandlung darüber hinausgehender Probleme.	Kann syntaktische Fehler stets und darüber hinausgehende Probleme häufig selbstständig lösen; findet z.T. selber Lösungshilfen und kann gegebene Hilfestellungen gut umsetzen.	Eigeninitiative und sehr selbstständige Lösung von auftretenden Schwierigkeiten; kreativer Umgang mit Problemen; Hilfestellungen werden selten und dann nur als Impuls benötigt.
Qualität der Beiträge	Fehlende Kenntnisse bzw. fachliche Defizite bei grundlegenden und aktuellen Themen.	Fachliche Ungenauigkeiten; unstrukturierte Beiträge; überwiegend reproduktive Beiträge.	Fachlich korrekte, gute, produktive Beiträge, die den Unterrichtsverlauf voranbringen, auch zu komplexeren Problemstellungen und Zusammenhängen.	Umfangreiche, produktive, sehr gute Beiträge, die neue Impulse für den Unterrichtsverlauf setzen, auch bei komplexen Problemstellungen und Transferaufgaben.

* **Werkzeuge:** Hierunter fallen u.a. die Darstellungsformen (Klassendiagramme, PAPs, Pseudocode, Programmcode), Basiskennnisse über Datentypen, Variablen und Methoden sowie die **Kontrollstrukturen** der Programmiersprache (Bedingte Anweisungen, logische Verknüpfungen, verschiedene Schleifentypen..)

1. Sequenz: Einstieg in die Informatik

Zeitraumen ca. 4 Wochen

Unterrichtssequenzen		Kompetenzen	Unterrichtliche Beispiele
<p>a) Information und Daten im geschichtlichen Kontext</p> <ul style="list-style-type: none"> Speichern von Daten mit informatischen Systemen am Beispiel der Schulrechner Vereinbarung von Richtlinien zur Datenspeicherung auf den Schulrechnern (z.B. Ordnerstruktur, Dateibezeichner etc.) <p>b) Informations- & Datenübermittlung in Netzen</p> <ul style="list-style-type: none"> Informatische Kommunikation in Rechnernetzen am Beispiel des Schulnetzwerks (z.B. Benutzeranmeldung, Netzwerkordner, Zugriffsrechte, Client-Server) Richtlinien zum verantwortungsvollen Umgang mit dem Internet <p>c) Aufbau informatischer Systeme (Aufbau und Funktionsweise eines Von-Neumann-Rechners)</p>		<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K). nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D). erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A). nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K). beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A). 	<p>Zu a): Arbeit mit der Schulplattform des Bistums Münster www.schulbistum.de (Rechteverwaltung)</p> <p>Zu b): Rechner zerlegen, Bauteile identifizieren, deren Aufgabe im Rechner recherchieren und zusammensetzen (mittels Fotos ein Wiki auf Schulbistum.de von verschiedenen Bauteilen erstellen)</p> <p>Zu a) und b): Aktueller Kontext (z.B. NSA), Comics, Cartoons ... (→ Geschichte der Informatik: Zeitstrahl entwickeln)</p> <p>Zu c): Aktuelle Fortbildungsreihe (Modul von-Neumann-Rechner)</p>
Binnendifferenzierung	Mögliche Eigenprägung	Möglicher Bezug zu anderen Fächern	
<p>Binnendifferenzierende Rechercharbeit</p> <p>Zu c) für starke Schülerinnen und Schüler: Exkurs maschinennahe Programmierung.</p>	<p>Zu a) und b): Verantwortlicher Umgang mit personenbezogenen Daten u. Rechten</p> <p>Zu a) und c): Kritische Bewertung der historischen Entwicklung von Informatiksystemen (Kriege als Motor z.B. von modernen Verschlüsselungstechniken)</p>	<p>Zu a) und c): Geschichte</p> <p>Zu b): Sozialwissenschaften</p> <p>Zu c): Physik</p>	<p>www.schulbistum.de</p> <p>Software: Johnny (Von-Neumann-Rechner)</p>

2. Sequenz: Grundlagen der objektorientierten Analyse

Zeitraumen ca. 4 Wochen

Unterrichtssequenzen		Kompetenzen	Unterrichtliche Beispiele
a) Benutzung einfacher Objekte b) Nutzung erster Java-Befehle (z.B. Erzeugen erster Objekte, Methodenaufruf mit und ohne Parameterübergabe) c) Klassenkarten / Objektkarten		Die Schülerinnen und Schüler... <ul style="list-style-type: none"> ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M). stellen den Zustand eines Objekts dar (D). stellen die Kommunikation zwischen Objekten grafisch dar (M). analysieren und erläutern eine objektorientierte Modellierung (A). dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D). ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M). 	<ul style="list-style-type: none"> Online-Lehrwerk: net-informatikbuch.de Projektarbeit in der Entwicklungsumgebung <i>BlueJ</i> unter Einbeziehung der 2D-Grafikbibliothek <i>Shapes & Sprites</i> Gloop Projekte (Fortbildungsmodul der Bezirksregierung)
Binnendifferenzierung	Mögliche Eigenprägung	Möglicher Bezug zu anderen Fächern	Verwendete Software / Medien
Differenzierende Aufgabenstellung innerhalb der durchgeführten Beispiele an jeder Stelle möglich.	Die Schülerinnen und Schüler arbeiten in Partnerarbeit zusammen und unterstützen sich gegenseitig beim Lernen.		net-informatikbuch.de BlueJ Shapes & Sprites

3. Sequenz: Grundlagen der objektorientierten Programmierung und algorithmischen Grundstrukturen in Java.

Zeitraumen ca. 8 Wochen

Unterrichtssequenzen		Kompetenzen	Unterrichtliche Beispiele	
a) Analyse einer gegebenen Sachsituation b) Identifikation von Objekten, Attributen und Methoden und deren Umsetzung im Quellcode c) Einführung von grundlegenden Datentypen und deren Basisoperationen d) Einführung grundlegender Kontrollstrukturen (Schleife, Verzweigung) mit deren PAPS oder Struktogrammen		Die Schülerinnen und Schüler... <ul style="list-style-type: none"> ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M). modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M). stellen die Kommunikation zwischen Objekten grafisch dar (D). dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D). implementieren Klassen in einer Programmiersprache (unter Nutzung dokumentierter Klassenbibliotheken) (I). entwerfen einfache Algorithmen (M). analysieren und erläutern einfache Algorithmen und Programme (A). modifizieren einfache Algorithmen und Programme (I). entwerfen einfache Algorithmen, stellen sie umgangssprachlich und grafisch dar (M). implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I). interpretieren Fehlermeldungen und korrigieren den Quellcode (I). 	<ul style="list-style-type: none"> BlueJ Link Universität Wuppertal: Material für den Anfangsunterricht in der Einführungsphase http://ddi.uni-wuppertal.de/material/materialsammlung/material/aktuelles.html 	
Binnendifferenzierung	Mögliche Eigenprägung	Möglicher Bezug zu anderen Fächern	Verwendete Software / Medien	
Differenzierende Aufgabenstellung innerhalb der durchgeführten Beispiele an jeder Stelle möglich.	Die Schülerinnen und Schüler arbeiten in Partnerarbeit zusammen und unterstützen sich gegenseitig beim Lernen.	Mathematik (Zufallszahlen, Runden)	net-informatikbuch.de BlueJ Shapes & Sprites	

4. Sequenz: Vertiefung der Objektorientierten Programmierung

Zeitraum ca. 4 Wochen

Unterrichtssequenzen		Kompetenzen	Unterrichtliche Beispiele
<p>Vertiefung der gelernten Begrifflichkeiten auf weitere Beispiele. Der Schwerpunkt liegt hier auf der Modellierung sowie den Entwurfs- und Implementationsdiagrammen.</p> <p>* eventuell GUI</p>		<p>Siehe auch unter 3., zusätzlich:</p> <ul style="list-style-type: none"> • modellieren Klassen unter Verwendung von Vererbung (M). • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M). • stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen (D). 	<ul style="list-style-type: none"> • Rollenspielprojekt in Anlehnung an Schöningh Informatik 2 • Shapes and Sprites (verschiedene Projekte)
Binnendifferenzierung	Mögliche Eigenprägung	Möglicher Bezug zu anderen Fächern	Verwendete Software / Medien
<p>Differenzierende Aufgabenstellung innerhalb der durchgeführten Beispiele an jeder Stelle möglich.</p> <p>* Für starke Schülerinnen und Schüler: Programmierung einer App mit Android SDK.</p>	<p>Die Schülerinnen und Schüler arbeiten in Partnerarbeit zusammen und unterstützen sich gegenseitig beim Lernen.</p>		<p>BlueJ UMLED JavaEditor (Android SDK)</p>

5. Sequenz: Kennenlernen einer höheren Datenstruktur

Zeitraumen ca. 8 Wochen

Unterrichtssequenzen		Kompetenzen	Unterrichtliche Beispiele	
<p>Einführung der Datenstruktur Array:</p> <p>a) Erzeugung von und Umgang mit Arrays</p> <p>b) Suchen und Sortieren auf Arrays</p> <p>c) Laufzeitanalyse von Sortierverfahren</p> <p>Bemerkung zu b) und c): Implementierung erst in der Stufe Q1 erforderlich.</p>		<p>Die Schülerinnen und Schüler...</p> <ul style="list-style-type: none"> analysieren und erläutern einfache Algorithmen und Programme (A). modifizieren einfache Algorithmen und Programme (I). entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M). analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D). entwerfen einen weiteren Algorithmus zum Sortieren (M). beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeitaufwand und Speicherplatzbedarf (A). 	<ul style="list-style-type: none"> Simulation von Lottoziehungen Spielbäume (Tic-Tac-Toe) 	
Binnendifferenzierung	Mögliche Eigenprägung	Möglicher Bezug zu anderen Fächern	Verwendete Software / Medien	
<p>Differenzierende Aufgabenstellung innerhalb der durchgeführten Beispiele an jeder Stelle möglich.</p> <p>*Für starke Schülerinnen und Schüler: Implementierung von b) und c) sowie vertiefende Laufzeitanalysen</p>	<p>Die Schülerinnen und Schüler arbeiten in Partnerarbeit zusammen und unterstützen sich gegenseitig beim Lernen.</p> <p>Kritischer Umgang mit Glücksspielen</p>	<p>Mathematik (Zufall und Wahrscheinlichkeit)</p>	<p>BlueJ</p>	

6. *Sequenz: Datensicherheit und Verschlüsselungsverfahren

Zeitraumen ca. 4 Wochen

Unterrichtssequenzen	Kompetenzen	Unterrichtliche Beispiele	
<p>a) String-Klasse und ausgewählte Methoden</p> <p>b) Einfaches Verschlüsselungsverfahren</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A). • erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A). • stellen ganze Zahlen und Zeichen in Binärcodes dar (D). • interpretieren Binärcodes als Zahlen und Zeichen (D). 	<p>Zu a): z.B. Palindrom-Test Galgenmännchen</p> <p>Zu b): z. B. Caesar-Verfahren Binärzahlen als Verschlüsselung</p>	
Binnendifferenzierung	Mögliche Eigenprägung	Möglicher Bezug zu anderen Fächern	Verwendete Software / Medien
<p>Differenzierende Aufgabenstellung innerhalb der durchgeführten Beispiele an jeder Stelle möglich.</p> <p>* Für starke Schülerinnen und Schüler: Implementierung höherer Verschlüsselungsverfahren über mehrdimensionale Arrays (Vigenère, Polybios)</p>	<p>Die Schülerinnen und Schüler arbeiten in Partnerarbeit zusammen und unterstützen sich gegenseitig beim Lernen.</p> <p>Auseinandersetzung mit unkritischem Versenden von Nachrichten.</p>	<p>Mathematik (Modulo-Rechnung, Dualzahlen)</p> <p>Physik (Schaltungen)</p>	<p>BlueJ</p>

7. *Sequenz: Abschlußprojekt

Zeitraumen ca. 4 Wochen

Unterrichtssequenzen		Kompetenzen	Unterrichtliche Beispiele
Komplexes Abschlussprojekt		Alle bisher erworbenen Kompetenzen werden vertieft.	<ul style="list-style-type: none"> • Spiel auf zweidimensionalen Arrays, z.B. „4 gewinnt“ • GUIs (Programmierung eines Taschenrechners)
Binnendifferenzierung	Mögliche Eigenprägung	Möglicher Bezug zu anderen Fächern	Verwendete Software / Medien
Für schwächere Schülerinnen und Schüler: Individuelle Wiederholung möglich.	Die Schülerinnen und Schüler arbeiten in Partnerarbeit zusammen und unterstützen sich gegenseitig beim Lernen.	div.	BlueJ JavaEditor Android SDK u.a.

1. Sequenz: Wiederholung und Vertiefung der Objektorientierten Programmierung

Zeitraumen ca. 6 Wochen

Unterrichtssequenzen		Kompetenzen	Unterrichtliche Beispiele
<p>1. Wiederholung und Erweiterung der objektorientierten Modellierung und Programmierung durch Analyse und Erweiterung eines kontextbezogenen Beispiels</p> <p>(a) Analyse der Problemstellung (b) Analyse und Entwicklung der Modellierung (Entwurfs- und Implementationsdiagramm) (c) Erweiterung der Modellierung im Implementationsdiagramm (Vererbung, abstrakte Klasse) (d) Kommunikation zwischen mindestens zwei Objekten (grafische Darstellung) (e) Dokumentation von Klassen (f) Implementierung der Anwendung oder von Teilen der Anwendung</p>		<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern objektorientierte Modellierungen (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), • ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen (D), • stellen die Kommunikation zwischen Objekten grafisch dar (D). 	<ul style="list-style-type: none"> • Phasen der objektorientierten Softwareentwicklung anhand konkreter Anforderungen • Verschiedene SAS Projekte mit dem Schwerpunkt Objektbeziehungen
Binnendifferenzierung	Mögliche Eigenprägung	Möglicher Bezug zu anderen Fächern	Verwendete Software / Medien
<ul style="list-style-type: none"> • Eigene GUI erstellen • Entwurfsdiagramme vorgeben • Teillösungen erstellen 	<p>Umgang mit den Besonderheiten von 3D-Umgebungen</p>	<p>Mathematik: Zufallszahlen Gesellschaftswissenschaftliche Bezüge</p>	<p>BlueJ, UMLED, JavaEditor</p>

2. Sequenz: Lineare Datenstrukturen

Zeitraumen ca. 12 Wochen

Unterrichtssequenzen		Kompetenzen	Unterrichtliche Beispiele	
<p>1. Die Datenstruktur Schlange und Stapel im Anwendungskontext unter Nutzung der Klassen Stack und Queue</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klassen Stack und Queue</p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klassen</p>		<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), • analysieren und erläutern Algorithmen und Programme (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • modifizieren Algorithmen und Programme (I), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen (I), stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D). 	<p>Zu 1)</p> <ul style="list-style-type: none"> • Wartezimmer (Queue) <p><i>Beispiel:</i> Patientenwarteschlange siehe schulinterner Lehrplan auf QUA-LiS NRW: Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.2 – Warteschlange</p> <ul style="list-style-type: none"> • UPN-Rechner oder Rangierbahnhof (Stack) <p>siehe schulinterner Lehrplan auf QUA-LiS NRW: Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1-II.3 – Anwendungen zu linearen Datenstrukturen</p>	
<p>2. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse List</p> <p>(a) Erarbeitung der Vorteile der Klasse List im Gegensatz zu den bereits bekannten linearen Strukturen</p> <p>(b) Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse List.</p>			<p>Zu 2 und 3.)</p> <ul style="list-style-type: none"> • Vokabeltrainer (Liste, Stack und Queue) <p>Siehe <i>Informatik II, Schöning</i>: In diesem Projekt können die drei Datenstrukturen gut miteinander kombiniert werden. Vokabeln werden in Listen verwaltet, um jederzeit Wörter entfernen oder hinzufügen zu können. Schon gelernte und noch nicht gekonnte Vokabeln werden in einer Queue verwaltet, so dass sie in regelmäßigen Abständen immer wieder abgefragt werden können.</p>	
<p>3. Vertiefung - Anwendungen von Listen, Stapeln oder Schlangen in mindestens einem weiteren Kontext</p>				
Binnendifferenzierung	Mögliche Eigenprägung	Möglicher Bezug zu anderen Fächern	Verwendete Software / Medien	
<p>Fördernd: Teillösungen zur Implementation</p> <p>Fordernd: Standardmethoden auf NRW-Datenstrukturen selbst implementieren</p>	<p>Hier sollte Aufgabenteilung bei Partner- bzw. Gruppenarbeiten im Vordergrund stehen: Die SuS unterstützen sich gegenseitig beim Lernen.</p>	<p>Terme in der Mathematik, kontextabhängige Bezüge</p>	<p>BlueJ, UMLLED, JavaEditor</p>	

3. Sequenz: Suchen und Sortieren auf linearen Datenstrukturen

Zeitraumen ca. 6 Wochen

Unterrichtssequenzen		Kompetenzen	Unterrichtliche Beispiele	
<p>1. Suchen von Daten in Listen und Arrays</p> <p>(a) Lineare Suche in Listen und in Arrays</p> <p>(b) Binäre Suche in Arrays als Beispiel für rekursives Problemlösen</p> <p>(c) Untersuchung der beiden Suchverfahren hinsichtlich ihrer Effizienz (Laufzeitverhalten, Speicherbedarf)</p> <p>2. Sortieren in Listen und Arrays - Entwicklung und Implementierung von iterativen und rekursiven Sortierverfahren</p> <p>(a) Entwicklung und Implementierung eines einfachen Sortierverfahrens für eine Liste</p> <p>(b) Implementierung eines einfachen Sortierverfahrens für ein Feld</p> <p>(c) Entwicklung eines rekursiven Sortierverfahrens für ein Feld (z.B. Sortieren durch Mischen)</p> <p>3. Untersuchung der Effizienz der Sortierverfahren „Sortieren durch direktes Einfügen“ und „Quicksort“ auf Feldern</p> <p>(a) Grafische Veranschaulichung der Sortierverfahren</p> <p>(b) Untersuchung der Anzahl der Vergleichsoperationen und des Speicherbedarf bei beiden Sortierverfahren Beurteilung der Effizienz der beiden Sortierverfahren</p>		<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> analysieren und erläutern Algorithmen und Programme (A), beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A), entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M), modifizieren Algorithmen und Programme (I), implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I), nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), interpretieren Fehlermeldungen und korrigieren den Quellcode (I), testen Programme systematisch anhand von Beispielen (I), stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D). 	<ul style="list-style-type: none"> Fortführung des Vorhabens „Wartezimmer“: Namensliste sortieren bzw. Sortieren nach diversen Kriterien Bundesjugendspiele siehe schulinterner Lehrplan auf QUA-LiS NRW: Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.3 Suchen und Sortieren Zeit- und Aufwandsmessprogramm für Sortieralgorithmen 	
Binnendifferenzierung	Mögliche Eigenprägung	Möglicher Bezug zu anderen Fächern	Verwendete Software / Medien	
<p>Aufwandsanalyse praktisch/theoretisch</p> <p>Referate zur O-Notation</p> <p>Weitere Sortierverfahren</p>	<ul style="list-style-type: none"> Relevanz von Such- und Sortierverfahren für die Gesellschaft „Ordnung ist das halbe Leben“: Moderne Lagerhaltung ohne den Überblick zu verlieren, auch im privaten und schulischen Bereich 	<p>Mathematische Beweise, insbesondere vollständige Induktion (z.B. $O(n^2)$ bei InsertSort)</p>	<p>Software: BlueJ, UMLed, JavaEditor</p> <p>Weitere Medien:</p> <ul style="list-style-type: none"> Programm zur Zeitmessung Demos zu den Sortierverfahren (Tänze, Filme erstellen) 	

4. Sequenz: Datenbanken

Zeitraumen ca. 8 Wochen

Unterrichtssequenzen		Kompetenzen	Unterrichtsvorhaben/Unterrichtliche Beispiele
<p>1. Nutzung von relationalen Datenbanken</p> <p>(a) Aufbau von Datenbanken und Grundbegriffe</p> <ul style="list-style-type: none"> Entwicklung von Fragestellungen zur vorhandenen DB Analyse der Struktur der vorgegebenen DB und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, DB-Schema <p>(b) SQL-Abfragen</p> <ul style="list-style-type: none"> Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL auf einer Tabelle Analyse und Erarbeitung von SQL-Abfragen auf einer oder mehreren Tabellen (Arithmetische- und Vergleichsoperatoren) <p>(c) Vertiefung an einem weiteren Datenbankbeispiel</p> <p>(d) Datenschutzaspekte, Zugriffsrechte</p> <p>2. Modellierung von relationalen Datenbanken</p> <p>(a) Entity-Relationship-Diagramm</p> <ul style="list-style-type: none"> Ermittlung von Entitäten, zugehörigen Attributen, Relationen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms Erläuterung und Modifizierung einer DB-Modellierung <p>(b) Entwicklung einer Datenbank aus einem Datenbankentwurf</p> <ul style="list-style-type: none"> Modellierung eines relationalen Datenbankschemas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln <p>(c) Redundanz, Konsistenz und Normalformen</p> <ul style="list-style-type: none"> Untersuchung einer Datenbank hinsichtlich Konsistenz und Redundanz in einer Anwendungssituation Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten) 		<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A), analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A), analysieren und erläutern eine Datenbankmodellierung (A), erläutern die Eigenschaften normalisierter Datenbankschemata (A), bestimmen Primär- und Sekundärschlüssel (M), ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M), modifizieren eine Datenbankmodellierung (M), modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M), bestimmen Primär- und Sekundärschlüssel (M), überführen Datenbankschemata in vorgegebene Normalformen (M), verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I), ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D), stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D), überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D). 	<ul style="list-style-type: none"> <i>Beispiel:</i> Streaming-Dienst VideoCenter ist die Simulation einer Online-Videothek für den Informatik-Unterricht mit Webfrontends zur Verwaltung der Kunden, der Videos und der Ausleihe. Außerdem ist es möglich direkt SQL-Abfragen einzugeben. Es ist auch möglich, die Datenbank herunter zu laden und lokal zu installieren. Unter http://dokumentation.videocenter.schule.de/old/video/index.html (abgerufen: 30.03.2014) findet man den Link zu dem VideoCenter-System sowie nähere Informationen. Lesenswert ist auch die dort verlinkte „Dokumentation der Fallstudie“ mit didaktischem Material, welches alternativ bzw. ergänzend zu der im Folgenden beschriebenen Durchführung verwendet werden kann. Artikelverwaltung: Script auf Schulbistum Aufgaben Klett, Band2
Binnendifferenzierung	Mögliche Eigenprägung	Mögl. Bz. z. and. Fächern	Verwendete Software / Medien
<ul style="list-style-type: none"> Datenbankprogrammierung (Trigger, Views, Procedures) Embedded SQL: Zugriff auf Datenbanken aus Java-Programmen 	<p>Umstrittene Datenbanken: Vorratsdatenhaltung Verantwortlicher Umgang mit personenbezogenen Daten und Rechten</p>	<p>Mathematik: Mengenlehre</p>	<p><u>Software:</u> SQLite, MySQL, MS Access oder LibreOffice Base</p>

5. Sequenz: Rekursive Datenstrukturen, Bäume

Zeitraumen ca. 8 Wochen

Unterrichtssequenzen	Kompetenzen	Unterrichtsvorhaben/Unterrichtliche Beispiele
<p>1. Analyse von Baumstrukturen in verschiedenen Kontexten</p> <p>(a) Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit)</p> <p>(b) Aufbau und Darstellung von binären Bäumen anhand von Baumstrukturen in verschiedenen Kontexten</p> <p>2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse BinaryTree</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms</p> <p>(c) Erarbeitung der Klasse BinaryTree und beispielhafte Anwendung der Operationen</p> <p>(d) Implementierung der Anwendung oder von Teilen der Anwendung</p> <p>(e) Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf</p> <p>3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse BinarySearchTree</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms, grafische Darstellung eines</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), • analysieren und erläutern Algorithmen und Programme (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D), stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D). 	<p>Zu 1) <i>Beispiel:</i> Ternbaum oder Ahnenbaum</p> <p><i>Weitere Beispiele für Anwendungskontexte für binäre Bäume:</i></p> <p><i>Beispiel:</i> Suchbäume (zur sortierten Speicherung von Daten) oder Entscheidungsbäume oder Codierungsbäume</p> <p>siehe schulinterner Lehrplan auf QUA-LiS NRW: Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Binärbaum</p> <p>Zu 2) <i>Beispiel:</i> Informatikerbaum als binärer Baum</p> <p>siehe schulinterner Lehrplan auf QUA-LiS NRW: Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Binärbaum</p> <p>Zu 3) <i>Beispiel:</i> Informatikerbaum als Suchbaum</p> <p>siehe schulinterner Lehrplan auf QUA-LiS NRW: Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Binärer Suchbaum</p> <p>Zu 4) <i>Beispiel:</i> Codierungsbäume, Huffman-Codierung, Buchindex, Entscheidungsbäume, Ternbaum oder Ahnenbaum (s.o.)</p>



<p>binären Suchbaums und Erarbeitung der Struktureigenschaften</p> <p>(c) Erarbeitung der Klasse BinarySearchTree und Einführung des Interface Item zur Realisierung einer geeigneten Ordnungsrelation</p> <p>(d) Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums</p> <p>4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen</p>		<p>siehe schulinterner Lehrplan auf QUA-LiS NRW: Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Anwendung Binärbaum</p> <p>Morsebaum (Implementationsmaterialen 2012)</p>	
Binnendifferenzierung	Mögliche Eigenprägung	Möglicher Bezug zu anderen Fächern	Verwendete Software / Medien
<p>Graphische Umsetzung Implementationen der NRW-Methoden AVL – Bäume</p>		<p>Mathematik: Terme Alle Fächer: Entscheidungsbäume</p>	<p>BlueJ</p>

6. Sequenz: Automaten und formale Sprachen

Zeitraumen ca. 10 Wochen

Unterrichtssequenzen		Kompetenzen	Unterrichtsvorhaben/Unterrichtliche Beispiele	
<p>1. Endliche Automaten</p> <p>(a) Vom Automaten in den Schülerinnen und Schülern bekannten Kontexten zur formalen Beschreibung eines endlichen Automaten</p> <p>(b) Untersuchung, Darstellung und Entwicklung endlicher Automaten</p> <p>2. Untersuchung und Entwicklung von Grammatiken regulärer Sprachen</p> <p>(a) Erarbeitung der formalen Darstellung regulärer Grammatiken</p> <p>(b) Untersuchung, Modifikation und Entwicklung von Grammatiken</p> <p>(c) Entwicklung von endlichen Automaten zum Erkennen regulärer Sprachen die durch Grammatiken gegeben werden</p> <p>(d) Entwicklung regulärer Grammatiken zu endlichen Automaten</p> <p>3. Grenzen endlicher Automaten</p> <p>4. Grenzen der Automatisierbarkeit</p> <p>a) Vorstellung des Halteproblems</p> <p>b) Unlösbarkeit des Halteproblems</p> <p>c) Beurteilung des Einsatzes von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen</p>		<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens auf bestimmte Eingaben (A), analysieren und erläutern Grammatiken regulärer Sprachen (A), zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungskontext (A), ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A), entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M), entwickeln zur akzeptierten Sprache eines Automaten die zugehörige Grammatik (M), entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M), modifizieren Grammatiken regulärer Sprachen (M), entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M), stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D), ermitteln die Sprache, die ein endlicher Automat akzeptiert (D). beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D). 	<p><i>Beispiele:</i> Cola-Automat, Geldspielautomat, Roboter, Zustandsänderung eines Objekts „Auto“, Akzeptor für bestimmte Zahlen, Akzeptor für Teilwörter in längeren Zeichenketten, Akzeptor für Terme</p> <p>siehe schulinterner Lehrplan auf QUA-LiS NRW: Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.2 – Endliche Automaten, Formale Sprachen</p> <p><i>Beispiele:</i> reguläre Grammatik für Wörter mit ungerader Parität, Grammatik für Wörter, die bestimmte Zahlen repräsentieren, Satzgliedergrammatik</p> <p><i>Materialien:</i> (s.o.)</p> <p><i>Beispiele:</i> Klammerausdrücke, $a^n b^n$ im Vergleich zu $(ab)^n$</p> <p>Spracherkennung</p> <p>Autokennzeichen, E-Mail-Adressen, Schaltjahre, Passwörter</p>	
Binnendifferenzierung	Mögliche Eigenprägung	Möglicher Bezug zu anderen Fächern	Verwendete Software / Medien	
Kellerautomaten Turingmaschine	Problematik der Automatisierung in der Gesellschaft, Automatisierung und Rationalisierung	diverse	<p><u>Software:</u> JFlap, Atocc, BlueJ</p> <p><u>Filme:</u> The Imitation game</p>	

7. Sequenz: Sicherheit und Datenschutz in Netzstrukturen

Zeitraumen ca. 4 Wochen

Unterrichtssequenzen	Kompetenzen	Unterrichtsvorhaben/Unterrichtliche Beispiele	
<p>1. Daten in Netzwerken und Sicherheitsaspekte in Netzen sowie beim Zugriff auf Datenbanken</p> <p>(a) Beschreibung eines Datenbankzugriffs im Netz anhand eines Anwendungskontextes und einer Client-Server-Struktur zur Klärung der Funktionsweise eines Datenbankzugriffs</p> <p>(b) Netztopologien als Grundlage von Client-Server-Strukturen und TCP/IP-Schichtenmodell als Beispiel für eine Paketübermittlung in einem Netz</p> <p>(c) Vertraulichkeit, Integrität, Authentizität in Netzwerken sowie symmetrische und asymmetrische kryptografische Verfahren (Cäsar-, Vigenère-, RSA-Verfahren) als Methoden Daten im Netz verschlüsselt zu übertragen</p> <p>2. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • beschreiben und erläutern Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A), • analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A), • untersuchen und bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen, die Sicherheit von Informatiksystemen sowie die Einhaltung der Datenschutzbestimmungen und des Urheberrechts (A), • untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A), • nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D). 	<ul style="list-style-type: none"> • Verschlüsselungsverfahren (Caesar und Vigenère) • Aufbau eines Chat-Servers • schulinterner Lehrplan auf QUA-LIS NRW: Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben, Verschlüsselung Q1.5 - Zugriff auf Daten in Netzwerken • Zu 2: datadealer.com 	
Binnendifferenzierung	Mögliche Eigenprägung	Möglicher Bezug zu anderen Fächern	Verwendete Software / Medien
<p>Implementation eigener Server- Client-Anwendungen</p> <p>Steganographie</p>	<p>Schulbistum.de</p> <p>Reflexion über eigenes Verhalten in Netzwerken</p>	<p>gesellschaftswissenschaftliche Bezüge</p>	<p><u>Software</u>: Filius</p> <p><u>Filme</u>: Citizen Four, Auswahl CCC,</p>

8. Sequenz: Von-Neumann-Rechnerstrukturen

Zeitraumen ca. 6 Wochen

Unterrichtssequenzen		Kompetenzen	Unterrichtsvorhaben/Unterrichtliche Beispiele	
1. Von-Neumann-Architektur und die Ausführung maschinennaher Programme a) prinzipieller Aufbau einer von Neumann-Architektur mit CPU, Rechenwerk, Steuerwerk, Register und Hauptspeicher b) einige maschinennahe Befehlen und ihre Repräsentation in einem Binär-Code, der in einem Register gespeichert werden kann c) Analyse und Erläuterung der Funktionsweise eines einfachen maschinennahen Programms d) Auswirkungen der Automatisierung auf die Gesellschaft		Die Schülerinnen und Schüler <ul style="list-style-type: none"> • erläutern die Ausführung eines einfachen maschinennahen Programms sowie die Datenspeicherung auf einer „Von-Neumann-Architektur“ (A), • untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A). 	<i>Beispiel:</i> Addition von 4 zu einer eingegebenen Zahl mit einem Rechnermodell siehe schulinterner Lehrplan auf QUA-LiS NRW: Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 –Von-Neumann-Architektur und maschinennahe Programmierung <i>Beispiel:</i> Halteproblem siehe schulinterner Lehrplan auf QUA-LiS NRW: Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 - Halteproblem	
Binnendifferenzierung		Mögliche Eigenprägung	Möglicher Bezug zu anderen Fächern	Verwendete Software / Medien
Andere Architekturen		Unterscheidung von Mensch und Maschine (Sinn des Lebens)	Mathematik (Dualzahlen)	<u>Software:</u> Johnny Simulator MOPS