

Schulinternen Lehrplan

**für die gymnasiale Oberstufe des
Reichenbach-Gymnasiums Ennepetal**

Informatik

(Stand: 31.03.2015)

INHALT

1	Aufgaben und Ziele des Faches	3
2	Kompetenzbereiche, Inhaltsfelder und Kompetenzerwartungen	4
2.1	Kompetenzbereiche und Inhaltsfelder des Faches	4
2.2	Kompetenzerwartungen und inhaltliche Schwerpunkte zum Ende der Einführungsphase	6
2.2.1	<i>Übersichtsraster Unterrichtsvorhaben</i>	7
2.2.2	<i>Konkretisierte Unterrichtsvorhaben</i>	10
2.3	Kompetenzerwartungen und inhaltliche Schwerpunkte zum Ende der Qualifikationsphase	26
2.3.1	<i>Übersichtsraster Unterrichtsvorhaben</i>	26
2.3.2	<i>Konkretisierungen der Unterrichtsvorhaben</i>	30
2.4	Gestaltung von Unterricht	58
3	Lernerfolgsüberprüfung und Leistungsbewertung	59
4	Abiturprüfung	62

1 Aufgaben und Ziele des Faches

Das Leben in der heutigen Informationsgesellschaft ist maßgeblich geprägt durch den verbreiteten Einsatz von Informations- und Kommunikationstechnologien sowohl im privaten als auch im beruflichen Bereich. Daher ist ein verantwortlicher Umgang mit diesen Technologien, den die Schülerinnen und Schüler im Unterricht des Fachs Informatik erlernen, wichtiger denn je. Zudem bauen sie ein Grundverständnis für eine Vielzahl von informatischen Systemen auf, indem sie diese modellieren oder selbst konstruieren. Bei der theoretischen Erarbeitung der Strukturen bieten sich dabei stets praxisnahe Beispiele aus der Welt der Schülerinnen und Schüler an, wodurch ihr erworbenes Wissen direkt anwendbar wird. Und fast beiläufig eignen sie sich eine ausgeprägte Medienkompetenz an, da im Fach Informatik beinahe jeder Arbeitsschritt wie das Recherchieren, Programmieren, Präsentieren oder Dokumentieren den Umgang mit medialen Werkzeugen abverlangt. Die genannten Aspekte belegen bereits das Potenzial des Fachs Informatik zur Förderung der individuellen Fachkompetenz.

Die Unterrichtsinhalte im Fach Informatik werden oft im Rahmen von Projektarbeiten erschlossen, bei denen die Schülerinnen und Schüler in einem ersten Schritt selbstständig und selbstverantwortlich lernen, aber im weiteren Verlauf sich auch in Teamfähigkeit üben müssen, um an ein (bestmögliches) Ergebnis zu gelangen. Zum Teil können der Kontext der Projektarbeit oder zumindest einige Aspekte selbst gewählt werden. Hierdurch können die Schülerinnen und Schüler verstärkt eigene Interessen einbringen und sich auf diese Weise mit den Unterrichtsinhalten in größerem Maße identifizieren. In diesem Zusammenhang ist ergänzend die Wahlmöglichkeit von Themenbereichen zu nennen, die den curricularen Vorgaben zu entnehmen sind. Auch sie gewährleistet eine Orientierung des Unterrichts an den Interessen der Schülerinnen und Schüler. Insgesamt trägt das Fach Informatik daher in vielfältiger Weise zur Persönlichkeitsentwicklung jedes Einzelnen bei.

Darüber hinaus fördert das Fach Informatik auch die Kreativität der Schülerinnen und Schüler, da sie zum gestalterischen, innovativen und produktorientierten Arbeiten angeregt werden. Die Erstellung von modernen Webseiten, funktionellen Benutzungsoberflächen oder ansprechenden Präsentationen sind nur einige Themenfelder, bei denen das Design eine nicht unwesentliche Rolle spielt. Zudem werden im Unterricht Informationssysteme der realen Welt abgebildet und befinden sich deshalb in einem stetigen Wandel. Die zu thematisierenden Technologien und Verfahren verlangen daher ein innovatives Denken, sowohl von den Lernenden als auch vom Lehrenden. Schließlich richtet auch das Fach Informatik vermehrt den Blick auf ein zu entwickelndes Endprodukt. So steht am Ende der Arbeitsprozesse z.B. eine Webseite, eine Schaltung oder eine Software.

2 Kompetenzbereiche, Inhaltsfelder und Kompetenzerwartungen

Die rechtliche Grundlage für die Themenwahl im Informatikunterricht der Sekundarstufe II bilden die Vorgaben des Ministeriums für Schule und Weiterbildung, Wissenschaft und Forschung des Landes Nordrhein-Westfalen. Hierzu gehören zum einen der Kernlehrplan (veröffentlicht 2014), in dem u.a. die allgemeinen Grundsätze hinsichtlich der didaktischen Konzepte, der Methodik und der Bewertung der Leistungen vorgegeben werden. Zum anderen werden in den Vorgaben für das jeweilige Jahr der Abiturprüfung spezifische Angaben über die zu behandelnden Unterrichtsinhalte festgelegt.

2.1 Kompetenzbereiche und Inhaltsfelder des Faches

Die für den Informatikunterricht obligatorischen Inhalte lassen sich in den folgenden Inhaltsfeldern zusammenfassen:

- Daten und ihre Strukturierung
- Algorithmen
- Formale Sprachen und Automaten
- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Im Rahmen der Erarbeitung der fachlichen Inhalte sollen mit dem oben beschriebenen Anwendungsbezug Kompetenzen der folgenden zentralen Bereiche bei den Schülerinnen und Schülern ausgebildet werden:

- **Argumentieren**
 - Erläutern, Begründen und Beurteilen in informatischen Sachzusammenhängen und Prozessen
 - Bewertung von Nutzen, Grenzen und Auswirkungen von Informatiksystemen
- **Modellieren**
 - Beschreibung und Analyse von Informatiksystemen
 - Strukturierte Zerlegung von Ausgangsproblemen und deren Vernetzung
- **Implementieren**
 - Umsetzung eines Modells in ein Informatiksystem
 - Programmieren, Evaluieren und Validieren von Modellbestandteilen
 - Nutzung von geeigneten Werkzeugen
- **Darstellen und Interpretieren**
 - Darstellungsformen für informatische Modelle
 - Einhaltung fachspezifischer Notationen
 - anwendungsbezogene Interpretation von Darstellungen
- **Kommunizieren und Kooperieren**

- Nutzung von Fachbegriffen und Entwicklung einer Fachsprache
- Sachadäquate Darstellung und Dokumentation von komplexen Informatiksystemen
- Nutzung geeigneter Werkzeuge zur Unterstützung der Kommunikation
- Kooperative Vorgehensweise zur Planung und Durchführung von prozessorientierten Projekten

Als allgemeines Programmierkonzept kommt dabei durchweg der **objektorientierte Ansatz** unter Verwendung plattformunabhängiger Klassenbibliotheken zum Einsatz. Die jahrgangsspezifischen Ausprägungen werden nachfolgend im Detail dargestellt. Als Programmierumgebung dient **Java**. Die Wahl basiert auf der freien Verfügbarkeit dieses Programmiersystems, der hohen Bedeutung der Sprache in der heutigen Welt der Informatik und der entsprechenden Vorgabe durch das Ministeriums für Schule und Weiterbildung, Wissenschaft und Forschung des Landes Nordrhein-Westfalen.

2.2 Kompetenzerwartungen und inhaltliche Schwerpunkte zum Ende der Einführungsphase

Die Darstellung der Unterrichtsvorhaben im schulinternen Lehrplan besitzt den Anspruch, sämtliche im Kernlehrplan angeführten Kompetenzen abzudecken. Dies entspricht der Verpflichtung jeder Lehrkraft, Schülerinnen und Schülern Lerngelegenheiten zu ermöglichen, so dass alle Kompetenzerwartungen des Kernlehrplans von ihnen erfüllt werden können.

Die entsprechende Umsetzung erfolgt auf zwei Ebenen: der Übersichts- und der Konkretisierungsebene.

Im „Übersichtsraster Unterrichtsvorhaben“ (Kapitel 2.2.1 und 2.3.1) wird die für alle Lehrerinnen und Lehrer gemäß Fachkonferenzbeschluss verbindliche Verteilung der Unterrichtsvorhaben dargestellt. Das Übersichtsraster dient dazu, den Kolleginnen und Kollegen einen schnellen Überblick über die Zuordnung der Unterrichtsvorhaben zu den einzelnen Jahrgangsstufen sowie den im Kernlehrplan genannten Kompetenzen, Inhaltsfeldern und inhaltlichen Schwerpunkten zu verschaffen. Der ausgewiesene Zeitbedarf versteht sich als grobe Orientierungsgröße, die nach Bedarf über- oder unterschritten werden kann. Um Freiraum für Vertiefungen, besondere Schülerinteressen, aktuelle Themen bzw. die Erfordernisse anderer besonderer Ereignisse (z.B. Praktika, Kursfahrten o.ä.) zu erhalten, wurden im Rahmen dieses schulinternen Lehrplans ca. 75 Prozent der Bruttounterrichtszeit verplant.

Während der Fachkonferenzbeschluss zum „Übersichtsraster Unterrichtsvorhaben“ zur Gewährleistung vergleichbarer Standards sowie zur Absicherung von Lerngruppenübertritten und Lehrkraftwechseln für alle Mitglieder der Fachkonferenz Bindekraft entfalten soll, beinhaltet die Ausweisung „konkretisierter Unterrichtsvorhaben“ (Kapitel 2.2.2 und 2.3.2) Beispiele und Materialien, die empfehlenden Charakter haben. Referendarinnen und Referendaren sowie neuen Kolleginnen und Kollegen dienen diese vor allem zur standardbezogenen Orientierung in der neuen Schule.

Da in den folgenden Unterrichtsvorhaben Inhalte in der Regel anhand von Problemstellungen in Anwendungskontexten bearbeitet werden, werden in einigen Unterrichtsvorhaben jeweils mehrere Inhaltsfelder angesprochen.

2.2.1 Übersichtsraster Unterrichtsvorhaben

Einführungsphase	
<p><u>Unterrichtsvorhaben E-I</u></p> <p>Thema: <i>Grundlagen der Funktionsweise von Informatiksystemen, der digitalen Datenverarbeitung und des Datenschutzes</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Informatiksysteme • Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Einzelrechner • Dateisystem • Internet • Einsatz von Informatiksystemen • Wirkungen der Automatisierung • Geschichte der automatischen Datenverarbeitung • Digitalisierung <p>Zeitbedarf: 12 Stunden</p>	<p><u>Unterrichtsvorhaben E-II</u></p> <p>Thema: <i>Grundlagen der objektorientierten Analyse, Modellierung und Implementierung anhand von statischen Grafikszenen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Daten und ihre Strukturierung • Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Objekte und Klassen • Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: 12 Stunden</p>

Einführungsphase

Unterrichtsvorhaben E-III

Thema:

Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen in Java anhand von einfachen Animationen

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Implementieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Daten und ihre Strukturierung
- Algorithmen
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Syntax und Semantik einer Programmiersprache
- Analyse, Entwurf und Implementierung einfacher Algorithmen

Zeitbedarf: 21 Stunden

Unterrichtsvorhaben E-IV

Thema:

Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand von grafischen Spielen und Simulationen

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Implementieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Daten und ihre Strukturierung
- Algorithmen
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Syntax und Semantik einer Programmiersprache
- Analyse, Entwurf und Implementierung einfacher Algorithmen

Zeitbedarf: 24 Stunden

Einführungsphase

Unterrichtsvorhaben E-V

Thema:

Such- und Sortieralgorithmen anhand kontextbezogener Beispiele

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Algorithmen

Inhaltliche Schwerpunkte:

- Algorithmen zum Suchen und Sortieren
- Analyse, Entwurf und Implementierung einfacher Algorithmen

Zeitbedarf: 6 Stunden

Summe Einführungsphase: 75

2.2.2 Konkretisierte Unterrichtsvorhaben

Im Folgenden sollen die im *Unterkapitel 2.2.1* aufgeführten Unterrichtsvorhaben konkretisiert werden.

Verbindliche Festlegungen der Fachkonferenz:

Die Fachkonferenz der Beispielschule hat Themen, Leitfragen und die Ausführungen unter der Überschrift Vorhabenbezogene Konkretisierung verbindlich vereinbart, ebenso die Sequenzierung der Unterrichtsvorhaben (erste Tabellenspalte) und die ausgewiesenen Kompetenzen (zweite Tabellenspalte). Alle Mitglieder der Fachkonferenz haben sich darauf verständigt, in ihrem Unterricht Lerngelegenheiten anzubieten, so dass Schülerinnen und Schüler diese Kompetenzen im Rahmen der festgelegten Unterrichtssequenzen erwerben oder vertiefen können.

Unterrichtliche Anregungen:

Die angeführten Beispiele, Medien und Materialien sind dagegen Vorschläge bzw. Hilfen für die Lehrkräfte der Beispielschule. In diesen Bereichen sind Abweichungen von den vorgeschlagenen Vorgehensweisen möglich.

Den inhaltlichen Schwerpunkt der Einführungsphase stellt die Einführung in die objektorientierte Programmierung dar. Diese erfolgt unter Verwendung des Konzepts „Stifte und Mäuse“ (SuM), welches eine umfassende und didaktisch aufbereitete Klassenbibliothek bereitstellt, die u.a. das programmgesteuerte Zeichnen unterstützt.

Eine ausführliche Darstellung der Themen und Projekte ist dem Buch *Informatik mit Java – Band I* von Bernard Schriek zu entnehmen.

Die folgenden Kompetenzen aus dem Bereich *Kommunizieren und Kooperieren* werden in allen Unterrichtsvorhaben der Einführungsphase vertieft und sollen aus Gründen der Lesbarkeit nicht in jedem Unterrichtsvorhaben separat aufgeführt werden:

Die Schülerinnen und Schüler

- verwenden Fachausdrücke bei der Kommunikation über informatische Sachverhalte (K),
- präsentieren Arbeitsabläufe und -ergebnisse (K),
- kommunizieren und kooperieren in Gruppen und in Partnerarbeit (K),
- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K).

Unterrichtsvorhaben EF-I

Thema: Grundlagen der Funktionsweise von Informatiksystemen, der digitalen Datenverarbeitung und des Datenschutzes

Leitfragen: *Womit beschäftigt sich die Wissenschaft der Informatik? Wie kann die in der Schule vorhandene informatische Ausstattung genutzt werden? Welche Entwicklung durchlief die moderne Datenverarbeitung und welche Auswirkungen ergeben sich insbesondere hinsichtlich neuer Anforderungen an den Datenschutz daraus?*

Vorhabenbezogene Konkretisierung:

Das erste Unterrichtsvorhaben stellt eine allgemeine Einführung in das Fach Informatik dar. Dabei ist zu berücksichtigen, dass für manche Schülerinnen und Schüler in der Einführungsphase der erste Kontakt mit dem Unterrichtsfach Informatik stattfindet, so dass zu Beginn Grundlagen des Fachs behandelt werden müssen.

Die nachfolgenden Themenbereiche werden in Kleingruppen bearbeitet und in Form von Plakat- oder Folienpräsentationen vorgestellt. Schülerinnen und Schüler sollen dabei mit Unterstützung des Lehrenden selbstständige Recherchen zu ihren Themen anstellen und auch eine sinnvolle Eingrenzung ihres Themas vornehmen.

Der erste Themenbereich bezieht sich auf die Möglichkeit der Kodierung in Form von Daten. Des Weiteren wird auf die Übertragung von Daten im Sinne des Sender-Empfänger-Modells eingegangen. Dabei wird eine überblickartige Vorstellung der Kommunikation von Rechnern in Netzwerken erarbeitet.

Ferner soll der grundlegende Aufbau eines Rechnersystems im Sinne der Von-Neumann-Architektur erarbeitet werden und mit dem grundlegenden Prinzip der Datenverarbeitung (Eingabe-Verarbeitung-Ausgabe) in Beziehung gesetzt werden.

Bei der Beschäftigung mit Datenkodierung, Datenübermittlung und Datenverarbeitung ist jeweils ein Bezug zur konkreten Nutzung der informatischen Ausstattung der Schule herzustellen. So wird in die verantwortungsvolle Nutzung dieser Systeme eingeführt.

Darüber hinaus sollen die Schülerinnen und Schüler selbstständig informatische Themenbereiche aus dem Kontext der Geschichte der Datenverarbeitung und insbesondere den daraus sich ergebenden Fragen des Datenschutzes bearbeiten.

Anschließend wird verstärkt auf den Aspekt des Datenschutzes eingegangen. Dazu wird das Bundesdatenschutzgesetz in Auszügen behandelt und auf schülernahe Beispielsituationen zur Anwendung gebracht. Dabei steht keine formale juristische Bewertung der Beispielsituationen im Vordergrund, die im Rahmen eines Informatikunterrichts auch nicht geleistet werden kann, sondern vielmehr eine persönliche Einschätzung von Fällen im Geiste des Datenschutzgesetzes.

Zeitbedarf: 9 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Selbstständige Erarbeitung von Themen durch die Schülerinnen und Schüler</p> <p>(a) Mögliche Themen zur Erarbeitung in Kleingruppen:</p> <ul style="list-style-type: none"> • „Eine kleine Geschichte der Digitalisierung: vom Morsen zum modernen Digitalcomputer“ • „Eine kleine Geschichte der Kryptographie: von Caesar zur Enigma“ • „Von Nullen, Einsen und mehr: Stellenwertsysteme und wie man mit ihnen rechnet“ • „Kodieren von Texten und Bildern: ASCII, RGB und mehr“ • „Auswirkungen der Digitalisierung: Veränderungen der Arbeitswelt und Datenschutz“ • „Informations- und Datenübermittlung in Netzen: Sender-Empfänger-Modell, Client-Server, der technischen Umsetzung (Netzwerkadresse, Protokoll), verantwortungsvollen Umgang mit dem Internet“ • „Aufbau informatischer Systeme: Von-Neumann-Architektur, Identifikation des EVA-Prinzips“ <p>(b) Vorstellung und Diskussion durch Schülerinnen und Schüler</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • beschreiben und erläutern den Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A), • bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A), • erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A), • nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D), • stellen ganze Zahlen und Zeichen in Binärcodes dar (D), • interpretieren Binärcodes als Zahlen und Zeichen (D), • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K). 	<p><i>Beispiel:</i> Textkodierung Kodierung und Dekodierung von Texten mit unbekanntem Zeichensätzen (z.B. Wingdings)</p> <p><i>Beispiel:</i> Bildkodierung Kodierung von Bildinformationen in Raster- und Vektorgrafiken</p> <p><i>Beispiel:</i> Rollenspiel zur Paketvermittlung im Internet Schülerinnen und Schüler übernehmen die Rollen von Clients und Routern. Sie schicken spielerisch Informationen auf Karten von einem Schüler-Client zum anderen. Jede Schülerin und jeder Schüler hat eine Adresse, jeder Router darüber hinaus eine Routingtabelle. Mit Hilfe der Tabelle und einem Würfel wird entschieden, wie ein Paket weiter vermittelt wird.</p> <p><i>Material:</i> Demonstrationshardware Durch Demontage eines Demonstrationsrechners entdecken Schülerinnen und Schüler die verschiedenen Hardwarekomponenten eines Informatiksystems. Als Demonstrationsrechner bietet sich ein ausrangierter Schulrechner an.</p>

2. Erarbeitung von Operationen zwischen Binärzahlen		
--	--	--

(a) Addition von Binärzahlen

(b) Darstellung von negativen Zahlen

(c) Subtraktion von Binärzahlen

Unterrichtsvorhaben EF-II

Thema: Grundlagen der objektorientierten Analyse, Modellierung und Implementierung anhand von statischen Grafikszenen

Leitfrage: *Wie lassen sich Gegenstandsbereiche informatisch modellieren und im Sinne einer Simulation informatisch realisieren?*

Vorhabenbezogene Konkretisierung:

Ein zentraler Bestandteil des Informatikunterrichts der Einführungsphase ist die Objektorientierte Programmierung. Dieses Unterrichtsvorhaben führt in die Grundlagen der Analyse, Modellierung und Implementierung in diesem Kontext ein.

Dazu werden zunächst konkrete Gegenstandsbereiche aus der Lebenswelt der Schülerinnen und Schüler analysiert und im Sinne des Objektorientierten Paradigmas strukturiert. Dabei werden die grundlegenden Begriffe der Objektorientierung und Modellierungswerkzeuge wie Objektkarten, Klassenkarten, Beziehungsdiagramme oder Klassendokumentationen eingeführt.

Im Anschluss wird mit der Realisierung erster Projekte mit Hilfe der didaktischen Programmierumgebung SuM begonnen. Die von der Bibliothek vorgegebenen Klassen werden von Schülerinnen und Schülern in Teilen analysiert und entsprechende Objekte anhand einfacher Problemstellungen erprobt. Dazu muss der grundlegende Aufbau einer Java-Klasse thematisiert und zwischen Deklaration, Initialisierung und Methodenaufrufen unterschieden werden.

Das Unterrichtsvorhaben schließt mit einem Projekt, das komplexere grafische Elemente beinhaltet, so dass die Schülerinnen und Schüler mehr als nur die Klasse erstellen müssen, welche die Szene als Ganzes darstellt. Elemente der Szene müssen zu sinnhaften eigenen Klassen zusammengefasst werden, die dann ihre eigenen Attribute und Dienste besitzen.

Da bei der Umsetzung dieser ersten Projekte konsequent auf die Verwendung von Kontrollstrukturen verzichtet wird und der Quellcode aus einer rein linearen Sequenz besteht, ist auf diese Weise eine Fokussierung auf die Grundlagen der Objektorientierung möglich, ohne dass algorithmische Probleme ablenken. Natürlich kann die Arbeit an diesen Projekten unmittelbar zum nächsten Unterrichtsvorhaben führen. Dort stehen unter anderem Kontrollstrukturen im Mittelpunkt.

Zeitbedarf: 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Identifikation von Objekten</p> <p>(a) Am Beispiel eines lebensweltnahen Beispiels werden Objekte im Sinne der Objektorientierten Modellierung eingeführt.</p> <p>(b) Objekte werden mit Objektkarten visualisiert und mit sinnvollen Attributen und „Fähigkeiten“, d.h. Methoden versehen.</p> <p>(c) Manche Objekte sind prinzipiell typgleich und werden so zu einer Objektsorte bzw. Objektklasse zusammengefasst.</p> <p>(d) Vertiefung: Modellierung weiterer Beispiele ähnlichen Musters</p>	<p>Die Schülerinnen und Schüler</p> <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 (M), • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), • dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D), • modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M) 	<p><i>Beispiel:</i> Vogelschwarm Schülerinnen und Schüler betrachten einen Vogelschwarm als Menge gleichartiger Objekte, die in einer Klasse mit Attributen und Methoden zusammengefasst werden können.</p>
<p>2. Analyse von Klassen didaktischer Lernumgebungen</p> <p>(a) Objektorientierte Programmierung als modularisiertes Vorgehen (Entwicklung von Problemlösungen auf Grundlage vorhandener Klassen)</p> <p>(b) Teilanalyse der Klassen der didaktischen Lernumgebungen SuM</p>		<p><i>Materialien:</i> Dokumentation der didaktischen Bibliothek SuM</p>
<p>3. Implementierung zweidimensionaler, statischer Szenen</p> <p>(a) Grundaufbau einer Java-Klasse</p> <p>(b) Konzeption einer Szene mit einfachen</p>		<p><i>Beispiel:</i> Hausbau Schülerinnen und Schüler erstellen ein Programm unter Verwendung vorgegebener grafischer Objektklassen (Kreis, Rechteck)</p>

<p>grafischen Objekten</p> <ul style="list-style-type: none"> (c) Deklaration und Initialisierung von Objekten (d) Methodenaufrufe mit Parameterübergabe zur Manipulation von Objekteigenschaften (z.B. Farbe, Position, Drehung) 		
<p>4. Modellierung komplexerer grafisch repräsentierbarer Objekte</p> <ul style="list-style-type: none"> (a) Modellierung eines Programms mit eigenen Klassen, die sich selbst mit Hilfe von einfachen Objekten zeichnen (b) Implementierung eigener Methoden mit und ohne Parameterübergabe (c) Realisierung von Zustandsvariablen (d) Thematisierung des Geheimnisprinzips und des Autonomitätsprinzips von Objekten 		

Unterrichtsvorhaben EF-III

Thema: Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen in Java anhand von einfachen Animationen

Leitfragen: *Wie lassen sich benutzergesteuerte Animationen und Simulationen realisieren?*

Vorhabenbezogene Konkretisierung:

Der Schwerpunkt dieses Unterrichtsvorhabens liegt auf der Entwicklung mehrerer Projekte, die durch Eingaben des Benutzers gesteuerte Animationen aufweisen. Zunächst wird ein Projekt bearbeitet, das in Anlehnung an das vorangegangene Unterrichtsvorhaben eine Szene darstellt, die lediglich aus Objekten besteht, zu denen das didaktische System Klassen vorgibt. Einzelne Objekte der Szene werden animiert, um ein einfaches Spiel zu realisieren oder die Szene optisch aufzuwerten. Für die Umsetzung dieses Projekts werden Kontrollstrukturen in Form von Schleifen und Verzweigungen benötigt und eingeführt.

Sind an einem solchen Beispiel im Schwerpunkt Schleifen und Verzweigungen eingeführt worden, sollen diese Konzepte an weiteren Beispielprojekten eingeübt werden. Dabei muss es sich nicht zwangsläufig um solche handeln, bei denen Kontrollstrukturen lediglich zur Animation verwendet werden.

Komplexere Assoziationsbeziehungen zwischen Klassen werden in diesem Unterrichtsvorhaben zunächst nicht behandelt. Sie stellen den Schwerpunkt des folgenden Vorhabens dar.

Zeitbedarf: 21 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Bewegungsanimationen am Beispiel einfacher grafischer Objekte</p> <p>(a) Kontinuierliche Verschiebung eines Objekts mit Hilfe einer Schleife (While-Schleife)</p> <p>(b) Tastatur-/Mausabfrage zur Realisierung einer Schleifenbedingung für eine Animationsschleife</p> <p>(c) Mehrstufige Animationen mit mehreren sequenziellen Schleifen</p> <p>(d) Berechnung von Abständen zwischen Objekten mit Hilfsvariablen</p> <p>(e) Meldungen zur Kollision zweier Objekte mit Hilfe von Abstandsberechnungen und Verzweigungen (IF-Anweisungen)</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern einfache Algorithmen und Programme (A), • entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M), • 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), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M), • ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • modifizieren einfache Algorithmen und Programme (I), • implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I), 	<p><i>Beispiel:</i> Wurfspiel Die Schülerinnen und Schüler realisieren mit Objekten der SuM-Umgebung ein Spiel, bei dem sich ein Pfeil über den Bildschirm bewegt und auf eine runde Zielscheibe geworfen werden soll.</p> <p><i>Beispiel:</i> Fahrendes Auto Die Schülerinnen und Schüler implementieren eine Klasse Auto, deren Objekte sich über den Bildschirm benutzergesteuert bewegen lassen.</p> <p><i>Beispiel:</i> Billard Die Schülerinnen und Schüler realisieren ein Billardspiel mit einer oder mehreren sich bewegenden Kugeln auf einem Tisch.</p>
<p>2. Anwendung der Kontrollstrukturen in einem weiteren Animationsbeispiel</p>		<p><i>Beispiel:</i> Freihandzeichnen Die Schülerinnen und Schüler erstellen ein einfaches Grafikprogramm, bei dem ein Stift über die Mausbewegung gesteuert wird.</p>

	<ul style="list-style-type: none">• implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),• testen Programme schrittweise anhand von Beispielen (I),• interpretieren Fehlermeldungen und korrigieren den Quellcode (I).	
--	--	--

Unterrichtsvorhaben EF-IV

Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand von grafischen Spielen und Simulationen

Leitfrage: *Wie lassen sich komplexere Datenflüsse und Beziehungen zwischen Objekten und Klassen realisieren?*

Vorhabenbezogene Konkretisierung:

Dieses Unterrichtsvorhaben beschäftigt sich im Schwerpunkt mit dem Aufbau komplexerer Objektbeziehungen. Während in vorangegangenen Unterrichtsvorhaben Objekte nur jeweils solchen Objekten Nachrichten schicken konnten, die sie selbst erstellt haben, soll in diesem Unterrichtsvorhaben diese hierarchische Struktur aufgebrochen werden.

Dazu bedarf es zunächst einer präzisen Unterscheidung zwischen Objektreferenzen und Objekten, so dass klar wird, dass Dienste eines Objektes von unterschiedlichen Objekten über unterschiedliche Referenzen in Anspruch genommen werden können. Auch der Aufbau solcher Objektbeziehungen muss thematisiert werden. Des Weiteren wird das Prinzip der Vererbung im objektorientierten Sinne angesprochen. Dazu werden die wichtigsten Varianten der Vererbung anhand von verschiedenen Projekten vorgestellt. Zunächst wird die Vererbung als Spezialisierung im Sinne einer einfachen Erweiterung einer Oberklasse vorgestellt. Darauf folgt ein Projekt, welches die Vererbung durch Generalisierung und den Aspekt der späten Bindung thematisiert. In diesem Zusammenhang ist das Prinzip der abstrakten Klasse vorzustellen. Modellierungen sollen in Form von Implementationsdiagrammen erstellt werden.

Zeitbedarf: 24 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Vertiefung des Referenzbegriffs und Einführung des Prinzips der dynamischen Referenzierung</p> <p>(a) Einführung der Objektselektion mit der Maus</p> <p>(b) Steuerung einfacher grafischer Objekte über eine Referenz <i>aktuell</i>, die jeweils durch eine Klickselektion mit der Maus auf ein neues Objekt gesetzt werden kann.</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> analysieren und erläutern eine objektorientierte Modellierung (A), stellen die Kommunikation zwischen Objekten grafisch dar (M), ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), 	<p><i>Beispiel:</i> Autokorso</p> <p>Die Schülerinnen und Schüler erweitern das Programm „Auto“ um die Möglichkeit der Auswahl des sich zu bewegenden Autos.</p>
<p>2. Erarbeitung einer Simulation mit grafischen Objekten, die sich durch unterschiedliche Ergänzungen voneinander unterscheiden (Vererbung durch Spezialisierung)</p> <p>(a) Analyse und Erläuterung einer Basisversion der grafischen Klasse</p> <p>(b) Realisierung von grafischen Erweiterungen zur Basisklasse mit und ohne Vererbung (Implementationsdiagramm und Quellcode)</p> <p>(c) Verallgemeinerung und Reflexion des Prinzips der Vererbung am Beispiel der Spezialisierung</p>	<ul style="list-style-type: none"> modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M), ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M), modellieren Klassen unter Verwendung von Vererbung (M), implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), 	<p><i>Beispiel:</i> Billard</p> <p>Die Schülerinnen und Schüler erweitern das Programm „Billard“ um die Klasse „Reibungskugel“, die sich realitätsnah verlangsamt.</p>

3. Entwicklung einer Simulation mit grafischen Objekten, die sich durch unterschiedliche Ausgestaltungen von Methoden unterscheiden (Vererbung durch Generalisierung)

- (a) Analyse und Realisierung verschiedener Klassen mit gleichartigen Attributen und Methoden
- (b) Erstellung einer gemeinsamen abstrakten Oberklasse
- (c) Reflexion des Prinzips der späten Bindung und der Polymorphie

- testen Programme schrittweise anhand von Beispielen (I),
- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),
- modifizieren einfache Algorithmen und Programme (I),
- modellieren und implementieren abstrakte Klassen unter Verwendung von Vererbung durch Generalisieren (M/I)
- stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D),
- dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D).

Beispiel: Figuren

Die Schülerinnen und Schüler entwickeln eine Simulation von verschiedenen grafischen Objekten (Kreis, Quadrat), die ausgewählt und anschließend verändert werden können.

Beispiel: Zug

Die Schülerinnen und Schüler modellieren und implementieren einen Zug bestehend aus Wagons unterschiedlicher Typen.

Unterrichtsvorhaben EF-V

Thema: Such- und Sortieralgorithmen anhand kontextbezogener Beispiele

Leitfragen: *Wie können Objekte bzw. Daten effizient sortiert werden, so dass eine schnelle Suche möglich wird?*

Vorhabenbezogene Konkretisierung:

Dieses Unterrichtsvorhaben beschäftigt sich mit der Erarbeitung von Such- und Sortieralgorithmen. Der Schwerpunkt des Vorhabens liegt dabei auf den Algorithmen selbst und nicht auf deren Implementierung in einer Programmiersprache, auf die in diesem Vorhaben vollständig verzichtet werden soll.

Zunächst erarbeiten die Schülerinnen und Schüler mögliche Einsatzszenarien für Such- und Sortieralgorithmen, um sich der Bedeutung einer effizienten Lösung dieser Probleme bewusst zu werden. Anschließend werden Strategien zur Sortierung mit Hilfe eines explorativen Spiels von den Schülerinnen und Schülern selbst erarbeitet und hinsichtlich der Anzahl notwendiger Vergleiche auf ihre Effizienz untersucht.

Daran anschließend werden die erarbeiteten Strategien systematisiert und im Pseudocode notiert. Die Schülerinnen und Schüler sollen auf diese Weise das *Sortieren durch Vertauschen*, das *Sortieren durch Auswählen* und mindestens einen weiteren Sortieralgorithmus, kennen lernen.

Des Weiteren soll das Prinzip der *binären Suche* behandelt und nach Effizienzgesichtspunkten untersucht werden.

Zeitbedarf: 6 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Explorative Erarbeitung eines Sortierverfahrens</p> <p>(a) Sortierprobleme im Kontext informatischer Systeme und im Alltag (z.B. Dateisortierung, Tabellenkalkulation, Telefonbuch, Bundesligatabelle, usw.)</p> <p>(b) Vergleich zweier Elemente als Grundlage eines Sortieralgorithmus</p> <p>(c) Erarbeitung eines Sortieralgorithmus durch die Schülerinnen und Schüler</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeit und Speicherplatzbedarf (A), • entwerfen einen weiteren Algorithmus zum Sortieren (M), • analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D). 	<p><i>Beispiel:</i> Sortieren mit Waage Die Schülerinnen und Schüler bekommen die Aufgabe, kleine, optisch identische Kunststoffbehälter aufsteigend nach ihrem Gewicht zu sortieren. Dazu steht ihnen eine Balkenwaage zur Verfügung, mit deren Hilfe sie das Gewicht zweier Behälter vergleichen können.</p> <p><i>Materialien:</i> Computer science unplugged – Sorting Algorithms, URL: www.csunplugged.org/sorting-algorithms abgerufen: 30. 03. 2014</p>
<p>2. Systematisierung von Algorithmen und Effizienzbetrachtungen</p> <p>(a) Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode (auf jeden Fall: Sortieren durch Vertauschen, Sortieren durch Auswählen)</p> <p>(b) Anwendung von Sortieralgorithmen auf verschiedene Beispiele</p> <p>(c) Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche</p> <p>(d) Variante des Sortierens durch Auswählen</p> <p>(e) Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des Speicherplatzbe-</p>		<p><i>Beispiele:</i> Sortieren durch Auswählen, Sortieren durch Vertauschen, Quicksort Quicksort ist als Beispiel für einen Algorithmus nach dem Prinzip <i>Teile und Herrsche</i> gut zu behandeln. Kenntnisse in rekursiver Programmierung sind nicht erforderlich, da eine Implementierung nicht angestrebt wird.</p> <p><i>Materialien:</i> Computer science unplugged – Sorting Algorithms, URL: www.csunplugged.org/sorting-algorithms abgerufen: 30. 03. 2014</p>

<p>darfs</p> <p>(f) Analyse des weiteren Sortieralgorithmus (sofern nicht in Sequenz 1 und 2 bereits geschehen)</p>		
<p>3. Binäre Suche auf sortierten Daten</p> <p>(a) Suchaufgaben im Alltag und im Kontext informatischer Systeme</p> <p>(b) Evtl. Simulationsspiel zum effizienten Suchen mit binärer Suche</p> <p>(c) Effizienzbetrachtungen zur binären Suche</p>		<p><i>Beispiel:</i> Simulationsspiel zur binären Suche nach Tischtennisbällen Mehrere Tischtennisbälle sind nummeriert, sortiert und unter Bechern verdeckt. Mit Hilfe der binären Suche kann sehr schnell ein bestimmter Tischtennisball gefunden werden.</p> <p><i>Materialien:</i> Computer science unplugged – Searching Algorithms, URL: www.csunplugged.org/searching-algorithms, abgerufen: 30. 03. 2014</p>

2.3 Kompetenzerwartungen und inhaltliche Schwerpunkte zum Ende der Qualifikationsphase

2.3.1 Übersichtsraster Unterrichtsvorhaben

Qualifikationsphase 1	
<p><u>Unterrichtsvorhaben Q1-I</u></p> <p>Thema: <i>Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Algorithmen • Formale Sprachen und Automaten • Daten und ihre Strukturierung • Informatiksysteme <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Objekte und Klassen • Analyse, Entwurf und Implementierung von Algorithmen • Syntax und Semantik einer Programmiersprache • Nutzung von Informatiksystemen <p>Zeitbedarf: 12 Stunden</p>	<p><u>Unterrichtsvorhaben Q1-II</u></p> <p>Thema: <i>Modellierung und Implementierung von Anwendungen mit linearen, insbesondere dynamischen Datenstrukturen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Objekte und Klassen • Analyse, Entwurf und Implementierung von Algorithmen • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: 24 Stunden</p>

Qualifikationsphase 1

Unterrichtsvorhaben Q1-III

Thema:

Suchen und Sortieren auf linearen Datenstrukturen auch unter Berücksichtigung des Rekursionsprinzips

Zentrale Kompetenzen:

- Argumentieren
- Implementieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Algorithmen
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

- Analyse, Entwurf und Implementierung von Algorithmen
- Algorithmen in ausgewählten informatischen Kontexten
- Syntax und Semantik einer Programmiersprache

Zeitbedarf: 12 Stunden

Unterrichtsvorhaben Q1-IV

Thema:

Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Implementieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Daten und ihre Strukturierung
- Algorithmen
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Analyse, Entwurf und Implementierung von Algorithmen
- Algorithmen in ausgewählten informatischen Kontexten
- Syntax und Semantik einer Programmiersprache

Zeitbedarf: 26 Stunden

Summe Qualifikationsphase 1: 74 Stunden

Qualifikationsphase 2

Unterrichtsvorhaben Q2-I

Thema:

Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Daten und ihre Strukturierung
- Algorithmen
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

- Datenbanken
- Algorithmen in ausgewählten informatischen Kontexten
- Syntax und Semantik einer Programmiersprache

Zeitbedarf: 20 Stunden

Unterrichtsvorhaben Q2-II

Thema:

Formale Sprachen und Automaten

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Endliche Automaten und formale Sprachen

Inhaltliche Schwerpunkte:

- Endliche Automaten
- Grammatiken regulärer Sprachen
- Möglichkeiten und Grenzen von Automaten und formalen Sprachen

Zeitbedarf: 20 Stunden

Qualifikationsphase 2

Unterrichtsvorhaben Q2-III

Thema:

Sicherheit und Datenschutz in Netz- und Datenbankstrukturen

Zentrale Kompetenzen:

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatiksysteme
- Algorithmen
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Einzelrechner und Rechnernetzwerke
- Sicherheit
- Nutzung von Informatiksystemen, Wirkungen der Automatisierung

Zeitbedarf: 10 Stunden

Unterrichtsvorhaben Q2-IV

Thema:

Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

Zentrale Kompetenzen:

- Argumentieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Einzelrechner und Rechnernetzwerke
- Grenzen der Automatisierung

Zeitbedarf: 6 Stunden

Summe Qualifikationsphase 2: 56 Stunden

2.3.2 Konkretisierungen der Unterrichtsvorhaben

Die folgenden Kompetenzen aus dem Bereich *Kommunizieren und Kooperieren* werden in allen Unterrichtsvorhaben der Qualifikationsphase vertieft und sollen aus Gründen der Lesbarkeit nicht in jedem Unterrichtsvorhaben separat aufgeführt werden:

Die Schülerinnen und Schüler

- verwenden die Fachsprache bei der Kommunikation über informatische Sachverhalte (K),
- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung von Dateien unter Berücksichtigung der Rechteverwaltung (K),
- organisieren und koordinieren kooperatives und eigenverantwortliches Arbeiten (K),
- strukturieren den Arbeitsprozess, vereinbaren Schnittstellen und führen Ergebnisse zusammen (K),
- beurteilen Arbeitsorganisation, Arbeitsabläufe und Ergebnisse (K),
- präsentieren Arbeitsabläufe und -ergebnisse adressatengerecht (K).

Darüber hinaus sollen den Schülerinnen und Schülern von Beginn der Qualifikationsphase an die Vorgaben des Zentralabiturs transparent gemacht werden. Hierzu dient u.a. die Einbeziehung von Abituraufgaben der vergangenen Jahre begleitend zu den jeweils behandelten Themengebieten.

Unterrichtsvorhaben Q1-I:

Thema: Wiederholung der objektorientierten Modellierung und Programmierung

Leitfragen: *Wie modelliert und implementiert man zu einer Problemstellung in einem geeigneten Anwendungskontext Java-Klassen inklusive ihrer Attribute, Methoden und Beziehungen? Wie kann man die Modellierung und die Funktionsweise der Anwendung grafisch darstellen?*

Vorhabenbezogenen Konkretisierung:

Zu einer Problemstellung in einem Anwendungskontext soll eine Java-Anwendung entwickelt werden. Die Problemstellung soll so gewählt sein, dass für diese Anwendung die Verwendung einer abstrakten Oberklasse als Generalisierung verschiedener Unterklassen sinnvoll erscheint und eine Klasse durch eine Unterklasse spezialisiert werden kann. Um die Aufgabe einzugrenzen, können (nach der ersten Problemanalyse) einige Teile (Modellierungen oder Teile von Java-Klassen) vorgegeben werden.

Die Schülerinnen und Schülern erläutern und modifizieren den ersten Entwurf und modellieren sowie implementieren weitere Klassen und Methoden für eine entsprechende Anwendung. Klassen und ihre Beziehungen werden in einem Implementationsdiagramm dargestellt. Dabei werden Sichtbarkeitsbereiche zugeordnet. Exemplarisch wird eine Klasse dokumentiert. Außerdem wird das Nachrichtenkonzept der objektorientierten Programmierung wiederholt.

Zeitbedarf: 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<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 der Modellierung (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 und der Bezeichnung der Bezugsobjekte (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 Implementie- 	<p><i>Beispiel:</i> Wetthuepfen Für ein Wetthüpfen zwischen einem Hasen, einem Hund und einem Vogel werden die Tiere gezeichnet. Alle Tiere springen wiederholt nach links. Die Höhe und Weite jedes Hüpfers ist zufällig. Evtl. marschieren sie anschließend hintereinander her.</p> <p>oder</p> <p><i>Beispiel:</i> Ampel Es werden eine Ampel u.a. mit den Klassen „Ampel“, „Leuchte“ (als Spezialisierung der Klasse „Kreis“) und „Gehäuse“ modelliert und grundlegende Funktionen implementiert.</p> <p><i>Beispiel:</i> Körper Verschiedene mathematische Körper werden hierarchisch modelliert, z.B. Ober-Oberklasse „Prisma“, Oberklasse „Quader“ und Unterklasse „Würfel“.</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.1-Wiederholung (Download Q1-I.1)</p>

	<p>rung und zum Test von Informatiksystemen an (I),</p> <ul style="list-style-type: none">• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),• stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D),• dokumentieren Klassen (D).	
--	---	--

Unterrichtsvorhaben Q1-II:

Thema: Modellierung und Implementierung von Anwendungen mit linearen, insbesondere dynamischen Datenstrukturen

Leitfrage: *Wie können (beliebig viele) linear angeordnete Daten im Anwendungskontext verwaltet werden?*

Vorhabenbezogene Konkretisierung:

Nach Analyse einer Problemstellung in einem geeigneten Anwendungskontext, in dem Daten in einem Datenfeld verwaltet werden, wird die Funktionsweise von Datenfeldern und deren Schwäche hinsichtlich der fehlenden Dynamik thematisiert. Hieraus erwächst der Bedarf der Schlangen-Struktur, deren Aufbau mit Fokus auf dem First-In-First-Out-Prinzip an einem Beispiel dargestellt und die Operationen der Klasse `Queue` erläutert werden. Anschließend werden für die Anwendung notwendige Klassen modelliert und implementiert. Eine Klasse für eine den Anforderungen der Anwendung entsprechende Oberfläche sowie die Klasse `Queue` wird dabei von der Lehrkraft vorgegeben. Anschließend wird die Anwendung modifiziert, um den Umgang mit der Datenstruktur zu üben. Anhand einer Anwendung, in der Daten nach dem Last-In-First-Out-Prinzip verwaltet werden, werden Unterschiede zwischen den Datenstrukturen Schlange und Stapel erarbeitet. Um einfacher an Objekte zu gelangen, die zwischen anderen gespeichert sind, wird die Klasse `List` eingeführt und in einem Anwendungskontext verwendet. In mindestens einem weiteren Anwendungskontext wird die Verwaltung von Daten in Schlangen, Stapeln oder Listen vertieft. Modellierungen werden dabei in Entwurfs- und Implementationsdiagrammen dargestellt.

Zeitbedarf: 24 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Die Datenstruktur Datenfeld im Anwendungskontext</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der typischen Operationen mit Datenfeldern</p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Datenfelder</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • erläutern Operationen 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 Datensammlungen zu (M), 	<p><i>Beispiel:</i> Zufallszahlen Speicherung von Ergebnissen eines Würfelwurfs und Implementierung verschiedene Operationen (z.B. arithmetisches Mittel)</p> <p><i>oder</i></p> <p><i>Beispiel:</i> Kalender Modellierung eines Kalender zur Eingabe von Terminen für einen ausgewählten Monat/Tag/Uhrzeit und deren Ausgabe</p>
<p>2. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse Queue</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse <code>Queue</code></p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse <code>Queue</code></p>	<ul style="list-style-type: none"> • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • modifizieren Algorithmen und Programme (I), • implementieren 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 	<p><i>Beispiel:</i> Patientenwarteschlange (jeder kennt seinen Nachfolger bzw. alternativ: seinen Vorgänger) Sobald ein Patient in einer Arztpraxis eintrifft, werden sein Name und seine Krankenkasse erfasst. Die Verwaltung der Patientenwarteschlange geschieht über eine Klasse, die hier als Wartezimmer bezeichnet wird. Wesentliche Operationen sind das „Hinzufügen“ eines Patienten und das „Entfernen“ eines Patienten, wenn er zur Behandlung gerufen wird. Die Simulationsanwendung stellt eine GUI zur Verfügung, legt ein Wartezimmer an und steuert die Abläufe. Wesentlicher Aspekt des Projektes ist die Modellierung des Wartezimmers mit Hilfe der Klasse <code>Queue</code>. Anschließend kann der Funktionsumfang der Anwendung erweitert werden: Patienten können sich zusätzlich in die Warteschlange zum Blutdruckmessen einreihen. Objekte werden von zwei Schlangen verwaltet.</p>

	<p>von Beispielen (I),</p> <ul style="list-style-type: none"> stellen lineare Strukturen grafisch dar und erläutern ihren Aufbau (D). 	<p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.2 – Warteschlange (Download Q1-II.1)</p>
<p>3. Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse <code>Stack</code></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse <code>Stack</code></p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse <code>Stack</code></p>		<p><i>Beispiel:</i> Heftstapel In einem Heftstapel soll das Heft einer Schülerin gefunden werden.</p> <p><i>oder</i></p> <p><i>Beispiel:</i> Containerschiff Container werden auf einem Schiff geladen (in Form von Stapeln) und gelöscht.</p> <p><i>oder</i></p> <p><i>Beispiel:</i> Türme von Hanoi Verschieden große Scheiben werden auf drei Stapel der Größe nach geordnet positioniert. Der gesamte Scheibenstapel soll unter Einhaltung der Größenbedingung verschoben werden.</p>
<p>4. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse <code>List</code></p> <p>(a) Erarbeitung der Vorteile der Klasse <code>List</code> im Gegensatz zu den bereits bekannten linearen Strukturen</p> <p>(b) Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse <code>List</code>.</p>		<p><i>Beispiel:</i> Adressbuch Verwaltung von Kontaktdaten in Adresslisten und Pflege unterschiedlicher Kontaktgruppen</p> <p><i>Beispiel:</i> Abfahrtslauf Bei einem Abfahrtslauf kommen die Skifahrer nacheinander an und werden nach ihrer Zeit in eine Rangliste eingeordnet. Diese Rangliste wird in einer Anzeige ausgegeben. Ankommende Abfahrer müssen an jeder Stelle der Struktur, nicht nur am Ende oder Anfang eingefügt werden können.</p>

		<p>Materialien: Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.2 - Listen (Download Q1-II.2)</p>
<p>5. Vertiefung - Anwendungen von Listen, Stapeln oder Schlangen in mindestens einem weiteren Kontext</p>		<p>Beispiel: Skispringen Ein Skispringen hat folgenden Ablauf: Nach dem Sprung erhält der Springer eine Punktzahl und wird nach dieser Punktzahl in eine Rangliste eingeordnet. Die besten 30 Springer qualifizieren sich für den zweiten Durchgang. Sie starten in umgekehrter Reihenfolge gegenüber der Platzierung auf der Rangliste. Nach dem Sprung erhält der Springer wiederum eine Punktzahl und wird nach der Gesamtpunktzahl aus beiden Durchgängen in die endgültige Rangliste eingeordnet.</p> <p>Beispiel: Terme in Postfix-Notation Die sog. UPN (<i>Umgekehrt-Polnische-Notation</i>) bzw. <i>Postfix-Notation</i> eines Terms setzt den Operator hinter die Operanden. Um einen Term aus der gewohnten Infixschreibweise in einen Term in UPN umzuwandeln oder um den Wert des Terms zu berechnen, kann ein Stack verwendet werden.</p> <p>Beispiel: Rangierbahnhof Auf einem Güterbahnhof gibt es drei Gleise, die nur zu einer Seite offen sind. Wagons können also von einer Seite auf das Gleis fahren und nur rückwärts wieder hinausfahren. Die Wagons tragen Nummern, wobei die Nummer jedoch erst eingesehen werden kann, wenn der Wagon der vorderste an der offenen Gleisseite ist. (Zwischen den Wagons herumzuturnen, um die anderen Wagonnummern zu lesen, wäre zu gefährlich.)</p>

		<p>Zunächst stehen alle Wagons unsortiert auf einem Gleis. Ziel ist es, alle Wagons in ein anderes Gleis zu fahren, so dass dort die Nummern der Wagons vom Gleisende aus aufsteigend in richtiger Reihenfolge sind. Zusätzlich zu diesen beiden Gleisen gibt es ein Abstellgleis, das zum Rangieren benutzt werden kann.</p> <p><i>Beispiel:</i> Autos an einer Ampel zur Zufahrtsregelung Es soll eine Ampel zur Zufahrtsregelung in Java simuliert werden. An einem geradlinigen, senkrecht von unten nach oben verlaufenden Straßenstück, das von Autos nur einspurig in eine Richtung befahren werden kann, ist ein Haltepunkt markiert, an dem die Ampel steht. Bei einem Klick auf eine Schaltfläche mit der Aufschrift „Heranfahren“ soll ein neues Auto an den Haltepunkt heranfahren bzw. bis an das letzte Auto, das vor dem Haltepunkt wartet. Grünphasen der Ampel werden durch einen Klick auf eine Schaltfläche mit der Aufschrift „Weiterfahren“ simuliert. In jeder Grünphase darf jeweils nur ein Auto weiterfahren. Die anderen Autos rücken nach.</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1-II.3 – Anwendungen für lineare Datenstrukturen (Download Q1-II.3)</p>
<p>Im Leistungskurs zusätzlich:</p> <p>Doppelt-verkettete Listen, auch im Anwendungskontext</p>		

Unterrichtsvorhaben Q1-III:

Thema: Suchen und Sortieren auf linearen Datenstrukturen auch unter Berücksichtigung des Rekursionsprinzips

Leitfrage: *Wie kann man gespeicherte Informationen günstig (wieder-)finden oder sortieren?*

Vorhabenbezogene Konkretisierung:

In einem Anwendungskontext werden zunächst Informationen in einer linearen Liste bzw. einem Feld gesucht. Hierzu werden Verfahren entwickelt und implementiert bzw. analysiert und erläutert, wobei neben einem iterativen auch ein rekursives Verfahren thematisiert wird und mindestens ein Verfahren selbst entwickelt und implementiert wird. Die verschiedenen Verfahren werden hinsichtlich Speicherbedarf und Zahl der Vergleichsoperationen miteinander verglichen.

Anschließend werden die bereits in der Einführungsphase erarbeiteten Sortierverfahren wiederholt und implementiert (ebenfalls für lineare Listen und Felder).

Abschließend werden verschiedene Sortierverfahren hinsichtlich der Anzahl der benötigten Vergleichsoperationen und des Speicherbedarfs beurteilt.

Im Leistungskurs soll ergänzend auch ein rekursives Sortierverfahren entwickelt werden. Die Implementationen von Quicksort sowie dem Sortieren durch Einfügen werden analysiert und erläutert. Falls diese Verfahren vorher schon entdeckt wurden, sollen sie hier wiedererkannt werden. Die rekursive Abarbeitung eines Methodenaufrufs von Quicksort wird grafisch dargestellt.

Zeitbedarf: 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<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>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 	<p><i>Beispiel:</i> Adressbuch</p> <p>Für ein Adressverwaltungsprogramm soll eine Methode zum Suchen einer Adresse geschrieben werden.</p> <p>oder</p> <p><i>Beispiel:</i> Bundesjugendspiele</p> <p>Die Teilnehmer an Bundesjugendspielen nehmen an drei Disziplinen teil und erreichen dort Punktzahlen. Diese werden in einer Wettkampfkarte eingetragen und an das Wettkampfbüro gegeben. Zur Vereinfachung sollte sich das Modell auf die drei Disziplinen „Lauf“, „Sprung“ und „Wurf“ beschränken.</p> <p>Im Wettkampfbüro wird das Ergebnis erstellt. Das Programm soll dafür zunächst den Besten einer Disziplin herausuchen können und später das gesamte Ergebnis nach gewissen Kriterien sortieren können.</p> <p><i>Materialien:</i></p> <p>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.3 - Suchen und Sortieren (Download Q1-III.1)</p>
<p>2. Exkurs: Rekursion als Problemlösestrategie</p> <p>(a) Erarbeitung des rekursiven Prinzips an verschiedenen Beispielen</p> <p>(b) Implementierung von rekursiven Algo-</p>	<ul style="list-style-type: none"> • testen Programme systematisch anhand 	<p><i>Beispiele:</i> Fakultät, Fibonacci jeweils iterativ und rekursiv</p> <p>oder</p> <p><i>Beispiel:</i> Selbstähnliche Bäume, Palindrome</p>

<p>rithmen aus dem Bereich der Mathematik und zur grafischen Darstellung</p>	<p>von Beispielen (I),</p>	
<p>3. Sortieren in Listen und Arrays - Wiederholung und Implementierung von iterativen und rekursiven Sortierverfahren</p> <p>(a) Wiederholung 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>	<ul style="list-style-type: none"> stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D). 	<p><i>Beispiel:</i> Sortieren von Zahlen in Feldern</p> <p>oder</p> <p><i>Beispiel:</i> Adressbuch (s.o.)</p>
<p>4. Untersuchung der Effizienz der Sortierverfahren</p> <p>(a) Untersuchung der Anzahl der Vergleichsoperationen und des Speicherbedarf bei beiden Sortierverfahren</p> <p>(b) Beurteilung der Effizienz der Sortierverfahren</p>		
<p>Im Leistungskurs zusätzlich:</p> <p>Erarbeitung und Implementierung des Sortierverfahrens „Quicksort“ inklusive Effizienz-Analyse</p>		

Unterrichtsvorhaben Q1-IV:

Thema: Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen

Leitfragen: *Wie können Daten im Anwendungskontext mit Hilfe binärer Baumstrukturen verwaltet werden? Wie kann dabei der rekursive Aufbau der Baumstruktur genutzt werden? Welche Vor- und Nachteile haben Suchbäume für die geordnete Verwaltung von Daten?*

Vorhabenbezogene Konkretisierung:

Anhand von Beispielen für Baumstrukturen werden grundlegende Begriffe eingeführt und der rekursive Aufbau binärer Bäume dargestellt. Anschließend werden für eine Problemstellung in einem der Anwendungskontexte Klassen modelliert und implementiert. Dabei werden die Operationen der Datenstruktur Binärbaum thematisiert und die entsprechende Klasse `BinaryTree` (der Materialien für das Zentralabitur in NRW) der Vorgaben für das Zentralabitur NRW verwendet. Klassen und ihre Beziehungen werden in Entwurfs- und Implementationsdiagrammen dargestellt. Die Funktionsweise von Methoden wird anhand grafischer Darstellungen von Binärbäumen erläutert.

Unter anderem sollen die verschiedenen Baumtraversierungen (Pre-, Post- und Inorder) implementiert werden. Unterschiede bezüglich der Möglichkeit, den Baum anhand der Ausgabe der Baum Inhalte via Pre-, In- oder Postorder-Traversierung zu rekonstruieren, werden dabei ebenfalls angesprochen, indem die fehlende Umkehrbarkeit der Zuordnung Binärbaum → Inorder-Ausgabe an einem Beispiel verdeutlicht wird.

Eine Tiefensuche wird verwendet, um einen in der Baumstruktur gespeicherten Inhalt zu suchen.

Zu einer Problemstellung in einem entsprechenden Anwendungskontext werden die Operationen der Datenstruktur Suchbaum thematisiert und unter der Verwendung der Klasse `BinarySearchTree` (der Materialien für das Zentralabitur in NRW) weitere Klassen oder Methoden in diesem Anwendungskontext modelliert und implementiert. Auch in diesem Kontext werden grafische Darstellungen der Bäume verwendet.

Die Verwendung von binären Bäumen und Suchbäumen wird anhand weiterer Problemstellungen oder anderen Kontexten weiter geübt.

Im Leistungskurs wird zusätzlich die Datenstruktur Graph erarbeitet. Hierbei wird insbesondere auf die Tiefensuche in Graphen und das Problem des kürzesten Wegs thematisiert.

Zeitbedarf: 26 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<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>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“ 	<p><i>Beispiel:</i> Ordnerstruktur Der Aufbau von Dateiverwaltungssystemen im Betriebssystem</p> <p>oder</p> <p><i>Beispiel:</i> Ahnenbaum Die binäre Baumstruktur ergibt sich daraus, dass jede Person genau einen Vater und eine Mutter hat.</p> <p><i>Weitere Beispiele für Anwendungskontexte für binäre Bäume:</i></p> <p><i>Beispiel:</i> Entscheidungsbäume Um eine Entscheidung zu treffen, werden mehrere Fragen mit ja oder nein beantwortet. Die Fragen, die möglich sind, wenn die Antwort auf eine Frage mit „ja“ beantwortet wird, befinden sich im linken Teilbaum, die Fragen, die möglich sind, wenn die Antwort „nein“ lautet, stehen im rechten Teilbaum.</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Binärbaum (Download Q2-I.1)</p>
<p>2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der</p>		<p><i>Beispiel:</i> Codierungsbäume für Codierungen, deren Alphabet aus genau zwei Zeichen be-</p>

<p>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 <code>BinaryTree</code> 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>und „Teilen und Herrschen“ (M),</p> <ul style="list-style-type: none"> • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • modifizieren Algorithmen und Programme (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), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D). 	<p>steht</p> <p>Morse hat Buchstaben als Folge von Punkten und Strichen codiert. Diese Codierungen können in einem Binärbaum dargestellt werden, so dass ein Übergang zum linken Teilbaum einem Punkt und ein Übergang zum rechten Teilbaum einem Strich entspricht. Wenn man im Gesamtbaum startet und durch Übergänge zu linken oder rechten Teilbäumen einen Pfad zum gewünschten Buchstaben sucht, erhält man die Morsecodierung des Buchstabens.</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Binärbaum (Download Q2-I.2)</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 Implementationsdiagramm, grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften</p> <p>(c) Erarbeitung der Klasse <code>BinarySearchTree</code> und Einführung des Interface <code>Item</code> zur Realisierung einer geeigneten Ordnungsrelation</p> <p>(d) Implementierung der Anwendung oder</p>		<p><i>Beispiel:</i> Informatikerbaum als <i>Suchbaum</i> In einem binären <i>Suchbaum</i> werden die Namen und die Geburtsdaten von Informatikern lexikographisch geordnet abgespeichert. Alle Namen, die nach dieser Ordnung vor dem Namen im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem Namen im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.)</p> <p>Folgende Funktionalitäten werden benötigt:</p> <ul style="list-style-type: none"> • Einfügen der Informatiker-Daten in den Baum • Suchen nach einem Informatiker über den Schlüssel Name • Ausgabe des kompletten Datenbestands in nach Namen sortierter Reihenfolge <p><i>Materialien:</i></p>

<p>von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums</p>		<p>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Binärer Suchbaum (Download Q2-I.3)</p>
<p>4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen</p>		<p><i>Beispiel:</i> Huffman-Codierung</p> <p>oder</p> <p><i>Beispiel:</i> Buchindex Es soll eine Anwendung entwickelt werden, die anhand von Stichworten und zugehörigen Seitenzahlen ein Stichwortregister erstellt. Da die Stichwörter bei der Analyse des Buches häufig gesucht werden müssen, werden sie in der Klasse <code>Buchindex</code> als Suchbaum (Objekt der Klasse <code>BinarySearchTree</code>) verwaltet. Alle Inhalte, die nach einer Ordnung vor dem Inhalt im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem Inhalt im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.)</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Anwendung Binärbaum (Download Q2-I.4)</p>

<p>Im Leistungskurs zusätzlich:</p> <p>5. Die Datenstruktur Graph im Anwendungskontext unter Verwendung der Klasse graph</p> <ul style="list-style-type: none">(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramm(c) Darstellung eines Graphen mit Hilfe von Adjazenzlisten/-matrizen und Diagrammen(d) Erarbeitung der Struktureigenschaften(e) Erarbeitung der Klasse graph(f) Implementierung der Anwendung oder von Teilen der Anwendung inklusive der Tiefensuche und des Problems des kürzesten Wegs (Algorithmus von Dijkstra)		<p><i>Beispiel:</i> Navigationssystem</p>
---	--	---

Unterrichtsvorhaben Q2-I:

Thema: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

Leitfragen: *Wie können Fragestellungen mit Hilfe einer Datenbank beantwortet werden? Wie entwickelt man selbst eine Datenbank für einen Anwendungskontext?*

Vorhabenbezogene Konkretisierung:

Ausgehend von einer vorhandenen Datenbank entwickeln Schülerinnen und Schüler für sie relevante Fragestellungen, die mit dem vorhandenen Datenbestand beantwortet werden sollen. Zur Beantwortung dieser Fragestellungen wird die vorgegebene Datenbank von den Schülerinnen und Schülern analysiert und die notwendigen Grundbegriffe für Datenbanksysteme sowie die erforderlichen SQL-Abfragen werden erarbeitet.

In anderen Anwendungskontexten müssen Datenbanken erst noch entwickelt werden, um Daten zu speichern und Informationen für die Beantwortung von möglicherweise auftretenden Fragen zur Verfügung zu stellen. Dafür ermitteln Schülerinnen und Schüler in den Anwendungssituationen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten und stellen diese in Entity-Relationship-Modellen dar. Entity-Relationship-Modelle werden interpretiert und erläutert, modifiziert und in Datenbankschemata überführt. Mit Hilfe von SQL-Anweisungen können anschließend im Kontext relevante Informationen aus der Datenbank extrahiert werden.

Ein Entity-Relationship-Diagramm kann auch verwendet werden, um die Entitäten inklusive ihrer Attribute und Relationen in einem vorgegebenen Datenbankschema darzustellen.

An einem Beispiel wird verdeutlicht, dass in Datenbanken Redundanzen unerwünscht sind und Konsistenz gewährleistet sein sollte. Die 1. bis 3. Normalform wird als Gütekriterium für Datenbankentwürfe eingeführt. Datenbankschemata werden hinsichtlich der 1. bis 3. Normalform untersucht und (soweit nötig) normalisiert.

Im Leistungskurs wird zusätzlich ein relationales Datenbankschema als Datenbank implementiert.

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<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 Datenbank Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, Datenbankschema <p>(b) SQL-Abfragen</p> <ul style="list-style-type: none"> Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL (<code>SELECT (DISTINCT) ...FROM, WHERE, AND, OR, NOT</code>) auf einer Tabelle Analyse und Erarbeitung von SQL-Abfragen auf einer und mehrerer Tabelle zur Beantwortung der Fragestellungen (<code>JOIN, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, <>, >, <, >=, <=, LIKE, BETWEEN, IN, IS NULL</code>) <p>(c) Vertiefung an einem weiteren Datenbankbeispiel</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> erläutern die Eigenschaften und den Aufbau von Datenbanksystemen (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 Datenbankabfra- 	<p><i>Beispiel:</i> VideoCenter 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.</p> <p><i>Beispiel:</i> Schulbuchausleihe Unter www.brd.nrw.de/lerntreffs/informatik/structure/material/sek2/datenbanken.php (abgerufen: 30. 03. 2014) wird eine Datenbank zur Verfügung gestellt, die Daten einer Schulbuch-Ausleihe enthält (über 1000 Entleiher, 200 Bücher mit mehreren tausend Exemplaren und viele Ausleihvorgänge). Die Datenbank kann in OpenOffice eingebunden werden.</p> <p><i>Beispiel:</i> SQL-Tutorial Unter www.schulserver.hessen.de/darmstadt/lichtenberg/SQLTutorial/</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 Datenbankmodellierung <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>gen über mehrere verknüpfte Tabellen (D),</p> <ul style="list-style-type: none"> • stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D), • überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D). 	<p><i>Beispiel: Fahrradverleih</i> Der Fahrradverleih <i>BTR (BikesToRent)</i> verleiht unterschiedliche Typen von Fahrrädern diverser Firmen an seine Kunden. Die Kunden sind bei <i>BTR</i> registriert (Name, Adresse, Telefon). <i>BTR</i> kennt von den Fahrradfirmen den Namen und die Telefonnummer. Kunden von <i>BTR</i> können CityBikes, Treckingräder und Mountainbikes ausleihen.</p> <p><i>Beispiel: Reederei</i> Die Datenverwaltung einer Reederei soll in einem Datenbanksystem umgesetzt werden. Ausgehend von der Modellierung soll mit Hilfe eines ER-Modells und eines Datenbankschemas dieser erste Entwurf normalisiert und in einem Datenbanksystem umgesetzt werden. Es schließen sich diverse SQL-Abfragen an, wobei auf die Relationalgebra eingegangen wird.</p> <p><i>Beispiel: Buchungssystem</i> In dem Online-Buchungssystem einer Schule können die Lehrer Medienräume, Beamer, Laptops, Kameras, usw. für einen bestimmten Zeitpunkt buchen, der durch Datum und die Schulstunde festgelegt ist. Dazu ist die Datenbank zu modellieren, ggf. zu normalisieren und im Datenbanksystem umzusetzen. Weiter sollen sinnvolle Abfragen entwickelt werden. Unter http://mrbs.sourceforge.net (abgerufen: 30.03. 2014) findet man ein freies Online-Buchungssystem inklusive Demo, an Hand derer</p>

		<p>man erläutern kann, worum es in dem Projekt geht.</p> <p><i>Beispiel: Schulverwaltung</i> In einer Software werden die Schulhalbjahre, Jahrgangsstufen, Kurse, Klassen, Schüler, Lehrer und Noten einer Schule verwaltet. Man kann dann ablesen, dass z.B. Schüler X von Lehrer Y im 2. Halbjahr des Schuljahrs 2011/2012 in der Jahrgangsstufe 9 im Differenzierungsbereich im Fach Informatik die Note „sehr gut“ erhalten hat. Dazu ist die Datenbank zu modellieren, ggf. zu normalisieren und im Datenbanksystem umzusetzen. Weiter sollen sinnvolle Abfragen entwickelt werden und das Thema Datenschutz besprochen werden.</p>
<p>Im Leistungskurs zusätzlich:</p> <p>Implementation eines Datenbankschemas im Anwendungskontext</p>		

Unterrichtsvorhaben Q2-II:

Thema: Formale Sprachen und Automaten

Leitfragen: *Wie kann man (endliche) Automaten genau beschreiben? Wie können endliche Automaten (in alltäglichen Kontexten oder zu informatischen Problemstellungen) modelliert werden? Wie können Sprachen durch Grammatiken beschrieben werden? Welche Zusammenhänge gibt es zwischen formalen Sprachen, endlichen Automaten und regulären Grammatiken?*

Vorhabenbezogene Konkretisierung:

Anhand kontextbezogener Beispiele werden endliche Automaten entwickelt, untersucht und modifiziert. Dabei werden verschiedene Darstellungsformen für endliche Automaten ineinander überführt und die akzeptierten Sprachen endlicher Automaten ermittelt. An einem Beispiel wird ein nichtdeterministischer Akzeptor eingeführt als Alternative gegenüber einem entsprechenden deterministischen Akzeptor.

Anhand kontextbezogener Beispiele werden Grammatiken regulärer Sprachen entwickelt, untersucht und modifiziert. Der Zusammenhang zwischen regulären Grammatiken und endlichen Automaten wird verdeutlicht durch die Entwicklung von allgemeinen Verfahren zur Erstellung einer regulären Grammatik für die Sprache eines gegebenen endlichen Automaten bzw. zur Entwicklung eines endlichen Automaten, der genau die Sprache einer gegebenen regulären Grammatik akzeptiert.

Auch andere Grammatiken werden untersucht, entwickelt oder modifiziert. An einem Beispiel werden die Grenzen endlicher Automaten ausgelotet.

Im Leistungskurs werden zusätzlich kontextfreie Sprachen behandelt. Außerdem werden für reguläre Sprachen Scanner, Parser und Interpreter modelliert und implementiert.

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<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>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 Anwendungszusammenhang auf (A), • ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A), • entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M), • 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 	<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><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.2 – Endliche Automaten, Formale Sprachen (Download Q2-II.1)</p>

	<p>die jeweils andere Darstellungsform (D),</p> <ul style="list-style-type: none"> • ermitteln die Sprache, die ein endlicher Automat akzeptiert (D). • beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D). 	
<p>Im Leistungskurs zusätzlich:</p> <ul style="list-style-type: none"> (a) Grammatiken kontextfreier Sprachen (b) Analyse und Erläuterung von Kellerautomaten (c) Modellierung von Scanner, Parser und Interpreter (d) Implementierung von Scanner, Parser und Interpreter 		

Unterrichtsvorhaben Q2-III:

Thema: Sicherheit und Datenschutz in Netz- und Datenbankstrukturen

Leitfragen: *Wie werden Daten in Netzwerken übermittelt? Was sollte man in Bezug auf die Sicherheit beachten?*

Vorhabenbezogene Konkretisierung:

Anschließend an das Unterrichtsvorhaben zum Thema Datenbanken werden der Datenbankzugriff aus dem Netz, Topologien von Netzwerken, eine Client-Server-Struktur, das TCP/IP-Schichtenmodell sowie Sicherheitsaspekte beim Zugriff auf Datenbanken und verschiedene symmetrische und asymmetrische kryptografische Verfahren analysiert und erläutert. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht runden das Unterrichtsvorhaben ab.

Im Leistungskurs werden zusätzlich Algorithmen und Methoden zur Kommunikation in Netzwerken behandelt.

Zeitbedarf: 10 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<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</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	<p><i>Materialien:</i></p> <p>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben, Verschlüsselung Q1.5 - Zugriff auf Daten in Netzwerken (Download Q1-V.1)</p>

<p>Verfahren (Cäsar-, Vigenère-, RSA-Verfahren) als Methoden Daten im Netz verschlüsselt zu übertragen</p>	<p>die Einhaltung der Datenschutzbestimmungen und des Urheberrechts (A),</p> <ul style="list-style-type: none"> • 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). 	
<p>2. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht</p>		<p><i>Materialien:</i> <i>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1 5 - Datenschutz beim Videocenter, Materialblatt-Datenschutzgesetz (Download Q1-V.2)</i></p>
<p>Im Leistungskurs zusätzlich:</p> <p>(a) Analyse, Entwicklung und Implementation von Algorithmen und Methoden zur Client-Server-Kommunikation unter Verwendung der Klassen <code>Connection</code>, <code>Client</code>, <code>Server</code></p> <p>(b) Prinzip der Nebenläufigkeit</p> <p>(c) Entwickeln und Erweitern von Protokollen zur Kommunikation in Client-Server-Netzwerken</p>		

Unterrichtsvorhaben Q2-IV:

Thema: Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

Leitfragen: *Was sind die strukturellen Hauptbestandteile eines Computers und wie kann man sich die Ausführung eines maschinenahen Programms mit diesen Komponenten vorstellen? Welche Möglichkeiten bieten Informatiksysteme und wo liegen ihre Grenzen?*

Vorhabenbezogene Konkretisierung:

Anhand einer von-Neumann-Architektur und einem maschinennahen Programm wird die prinzipielle Arbeitsweise von Computern verdeutlicht.

Ausgehend von den prinzipiellen Grenzen endlicher Automaten liegt die Frage nach den Grenzen von Computern bzw. nach Grenzen der Automatisierbarkeit nahe. Mit Hilfe einer entsprechenden Java-Methode wird plausibel, dass es unmöglich ist, ein Informatiksystem zu entwickeln, das für jedes beliebige Computerprogramm und jede beliebige Eingabe entscheidet ob das Programm mit der Eingabe terminiert oder nicht (Halteproblem). Anschließend werden Vor- und Nachteile der Grenzen der Automatisierbarkeit angesprochen und der Einsatz von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen beurteilt.

Zeitbedarf: 6 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p>1. Von-Neumann-Architektur und die Ausführung maschinennaher Programme</p> <p>a) prinzipieller Aufbau einer von Neumann-Architektur mit CPU, Rechenwerk, Steuerwerk, Register und Hauptspeicher</p> <p>b) einige maschinennahe Befehle und ihre Repräsentation in einem Binär-Code, der in einem Register gespeichert werden kann</p> <p>c) Analyse und Erläuterung der Funktionsweise eines einfachen maschinennahen Programms</p>	<p>Die Schülerinnen und Schüler</p> <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). 	<p><i>Beispiel:</i> Addition von 4 zu einer eingegebenen Zahl mit einem Rechnermodell</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 –Von-Neumann-Architektur und maschinennahe Programmierung (Download Q2-III.1)</p>
<p>2. 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><i>Beispiel:</i> Halteproblem</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 - Halteproblem (Download Q2-III.2)</p>

2.4 Gestaltung von Unterricht

Der Unterricht in Informatik ist derart zu gestalten, dass dieser den Richtlinien folgend sowohl eine wissenschaftspropädeutische Ausbildung der Schülerinnen und Schüler sichert als auch die persönliche Entfaltung in sozialer Verantwortung ermöglicht. Die Organisation des Unterrichts soll demnach im Wesentlichen folgende Aspekte im Blick haben:

- Berücksichtigung der individuellen Schülerpersönlichkeit mit ihren Vorerfahrungen, Fähigkeiten und Leistungsvermögen
- Aktivierung der Schülerinnen und Schüler im Lernprozess
- Unterstützung kooperativer Arbeitsformen
- Schaffung von Lernarrangements zur Kompetenzförderung sowie Konstruktion von Wissen und dessen Sicherung
- Differenzierung in den Aufgabenstellungen
- Auswahl von Aufgaben mit Anwendungsbezug und Transfermöglichkeit

Bei der Planung von Unterricht ist grundsätzlich davon auszugehen, dass die Teilnehmer der jeweiligen Lerngruppe heterogene Lernvoraussetzungen aufweisen. Dies gilt insbesondere für die Einführungsphase, da hier Schülerinnen und Schüler ohne spezielles Vorwissen mit Lernenden, die in der Sekundarstufe I den Differenzierungskurs Informatik absolviert haben, gemeinsam unterrichtet werden. Dies erfordert differenzierende Zugänge zu den fachlichen Inhalten, die vorhandene Kenntnisse für den Unterricht nutzen, aber nicht voraussetzen.

Hinsichtlich des Einsatzes von kooperativen Arbeitsformen soll verstärkt auf das selbstständige und selbstorganisierte Arbeiten der Schülerinnen und Schüler Wert gelegt werden. Dazu bietet sich insbesondere die offene Arbeitsform der Projektarbeit an. Mit Blick auf die Qualifikationsphase und die Arbeitsweise bei professionellen Hard- und Softwareprojekten sollen die Schülerinnen und Schüler eine entsprechende Methodenkompetenz aufbauen.

Im Zentrum der Projektarbeit stehen dabei neben der Selbstständigkeit das wissenschafts-orientierte Arbeiten im Team ein fächerübergreifendes Thema und die Schulung der Dokumentations- und Präsentationsfähigkeit. Ausgangspunkt des Projekts ist stets eine Problemstellung, die in Form eines Projektplans mit konkreten Ziel- und Zeitvereinbarungen schriftlich fixiert wird. Bei der Durchführung des Projekts sollte auf eine arbeitsteilige Bearbeitung geachtet werden, wobei jeder Lernende am Ende der Einführungsphase über die angestrebte Kompetenz in allen Aufgabenbereichen verfügen sollte. Nach eventuell anfänglicher Steuerung des Arbeitsprozesses durch die Lehrkraft ist die Selbstorganisation durch das Team anzustreben.

Wesentlicher Bestandteil der Projektarbeit ist stets eine ausführliche Projektdokumentation, die begleitend zum Arbeitsprozess zu erstellen ist. Diese ist neben dem Endprodukt und der Schlusspräsentation Grundlage für eine Bewertung der Projektarbeit.

3 Lernerfolgsüberprüfung und Leistungsbewertung

Die Grundsätze der Leistungsbewertung ergeben sich aus den entsprechenden Bestimmungen der Allgemeinen Schulordnung (§§ 21 bis 23). Dabei wird Leistungsbewertung als ein kontinuierlicher Prozess verstanden, welcher sich auf die im Unterricht vermittelten Kenntnisse, Fähigkeiten und Fertigkeiten (siehe Kapitel 1) bezieht. Bewertet werden alle von Schülerinnen und Schülern im Zusammenhang mit dem Unterricht erbrachten Leistungen. Diese sind in Kapitel 2.2 und 2.3 näher ausgeführt.

Bei der Leistungsbewertung ist zudem auf folgende Aspekte zu achten:

- Die Kriterien der Leistungsbewertung müssen den Schülerinnen und Schülern transparent sein.
- Die der Leistungsbewertung zugrunde liegenden Anforderungen orientieren sich an den im Lehrplan formulierten Anforderungsbereichen.

3.1 Beurteilungsbereich Klausuren

Klausuren dienen der schriftlichen Überprüfung der Lernergebnisse in einem Kursabschnitt. Die Anzahl und Dauer der Klausuren richtet sich dabei nach den Vorgaben aus der APO-GOST und kann der nachfolgenden Tabelle entnommen werden:

	Anzahl	Dauer	
		Grundkurs	Leistungskurs
EF – 1. Halbjahr	1	2	
EF – 2. Halbjahr	1	2	
Q1 – 1. Halbjahr	2	2	3
Q1 – 2. Halbjahr	2	2	3
Q2 – 1. Halbjahr	2	3	4
Q2 – 2. Halbjahr	1	3	5

Im zweiten Halbjahr der Jahrgangsstufe Q1 kann dabei eine Klausur gleichwertig durch eine Facharbeit ersetzt werden.

Bei der Auswahl und Formulierung der Aufgabenstellung der Klausuren sollte darauf geachtet werden, dass die Schülerinnen und Schüler diese auf Basis der im Unterricht erworbenen Kenntnisse verstehen und selbstständig bearbeiten können. Um die Intentionen der Aufgaben transparent zu gestalten und eine adäquate Vorbereitung auf die Abiturprüfungen zu gewährleisten, ist die Verwendung von Operatoren bei der Aufgabenstellung zu empfehlen. Eine entsprechende Liste der Operatoren mit Zuordnung zu den Anforderungsbereichen ist auf der Homepage des Schulministeriums zu finden. Es bietet sich daher an, bereits im Rahmen des Unterrichts entspre-

chende Formulierungen zu wählen. Die Aufgabentypen, sowie die Anforderungsbereiche I-III sind entsprechend den Vorgaben in Kapitel 3 des Kernlehrplans zu beachten.

Die Bewertung der schriftlichen Leistungen in Klausuren erfolgt über ein Raster mit Hilfspunkten, die im Erwartungshorizont den einzelnen Kriterien zugeordnet sind.

Spätestens ab der Qualifikationsphase orientiert sich die Zuordnung der Hilfspunktsumme zu den Notenstufen an dem Zuordnungsschema des Zentralabiturs.

Von diesem kann aber im Einzelfall begründet abgewichen werden, wenn sich z.B. besonders originelle Teillösungen nicht durch Hilfspunkte gemäß den Kriterien des Erwartungshorizontes abbilden lassen oder eine Abwertung wegen besonders schwacher Darstellung (APO-GOST §13 (2)) angemessen erscheint.

3.2 Beurteilungsbereich Sonstige Mitarbeit

Der Beurteilungsbereich „Sonstige Leistungen“ umfasst alle Leistungen, die eine Schülerin bzw. ein Schüler im Zusammenhang mit dem Unterricht mit Ausnahme der Klausuren und ggf. der Facharbeit erbringt. Der Beurteilungsbereich ist dabei grundsätzlich gleichwertig mit dem Beurteilungsbereich „Klausuren“.

Zum Beurteilungsbereich „Sonstige Leistungen“ gehören Beiträge zum Unterrichtsgespräch, Leistungen in Hausaufgaben, Referaten, Protokollen und Präsentationen sowie insbesondere die Mitarbeit in Projekten. Die der Bewertung der vorgenannten Leistungen zugrunde liegenden Kriterien sind in der nachfolgenden Tabelle aufgeführt:

N o t e	Unterrichtsgespräch	Partner- / Gruppenarbeit / praktische Übungen im Programmieren/ Soft- wareprojekte/ sonstige Arbeitsformen
1	<ul style="list-style-type: none"> • beeinflusst durch umfassende und gut strukturierte Lösungsvorschläge entscheidend das Unterrichtsgespräch • bringt eigenständige Beiträge zu komplexen Sachverhalten • erfasst komplexe informatische Zusammenhänge und kann früher Gelerntes in informatischen Modellierungen einbringen 	<ul style="list-style-type: none"> • wirkt entscheidend bei der Planung und Durchführung mit • kann besondere Kenntnisse und algorithmische Ideen einbringen • kann Lösungswege und -methoden umfassend, sicher und gut verständlich darstellen
2	<ul style="list-style-type: none"> • gestaltet bei anspruchsvollen Problemstellungen das Unterrichtsgespräch mit eigenen Ideen • kann schwierige Sachverhalte verständlich darstellen • kann früher Gelerntes korrekt darstellen 	<ul style="list-style-type: none"> • wirkt aktiv bei der Planung und Durchführung mit • bringt Kenntnisse ein, die für das Arbeitsergebnis wichtig sind • kann Lösungswege und -methoden vollständig, richtig und verständlich darstellen
3	<ul style="list-style-type: none"> • beteiligt sich regelmäßig mit gehaltvollen Beiträgen • bringt zu grundlegenden Problemstellungen Lösungsansätze ein • kann den Unterrichtsstoff in das Reihenthema einordnen 	<ul style="list-style-type: none"> • beteiligt sich an der Planung und Durchführung • bringt Kenntnisse ein, die für die Lösung wichtig sind • stellt Lösungswege und -methoden in den wesentlichen Punkten richtig und nachvollziehbar dar
4	<ul style="list-style-type: none"> • beteiligt sich nicht immer am Unterrichtsgespräch • Beiträge sind überwiegend Antworten auf einfache und reproduktive Fragen • kann auf Aufforderung den Gegenstand der aktuellen Stunde im Wesentlichen in den Themenzusammenhang einordnen 	<ul style="list-style-type: none"> • beteiligt sich an der Lösung der Aufgaben, bearbeitet die Arbeitsaufträge • bringt Kenntnisse ein • kann Lösungswege und -methoden in den Grundzügen richtig darstellen
5	<ul style="list-style-type: none"> • beteiligt sich fast nie und ist unaufmerksam • kann kaum Beiträge einbringen • kann auf Aufforderung grundlegende Inhalte nicht oder nur falsch wiedergeben 	<ul style="list-style-type: none"> • bearbeitet nur langsam und wenig erfolgreich die Arbeitsaufträge • bringt keine Kenntnisse ein • kann Lösungswege und -methoden in den Grundzügen nicht richtig darstellen
6	<ul style="list-style-type: none"> • folgt dem Unterricht nicht • verweigert die Mitarbeit • Beiträge sind fast immer falsch 	<ul style="list-style-type: none"> • bearbeitet die Aufgabenstellungen nicht • kann Fragen nach Lösungen und Lösungswegen nicht beantworten

4 Abiturprüfung

Zur Abiturprüfung und den entsprechenden Vorgaben wird auf den Kernlehrplan bzw. die Allgemeine Prüfungsordnung für die Gymnasiale Oberstufe verwiesen.