



Schulinternes Curriculum

– Informatik –

Sekundarstufe II

Inhalt

1	Die Fachgruppe Informatik am Gymnasium am Geroweier	3
2	Entscheidungen zum Unterricht	4
2.1	Unterrichtsvorhaben	6
2.1.1	Konkretisierte Unterrichtsvorhaben EF	6
2.1.2	Konkretisierte Unterrichtsvorhaben Q1	24
2.1.3	Konkretisierte Unterrichtsvorhaben Q2	35
2.2	Grundsätze der fachmethodischen und fachdidaktischen Arbeit	45
2.3	Grundsätze der Leistungsbewertung und Leistungsrückmeldung	46
3	Entscheidungen zu fach- und unterrichtsübergreifenden Fragen	50
4	Qualitätssicherung und Evaluation	51

1 Die Fachgruppe Informatik am Gymnasium am Geroweier Mönchengladbach

Das Fach Informatik wird am Gymnasium am Geroweier in den Jahrgangsstufen 5 und 6 in jeweils einem Halbjahr zweistündig unterrichtet. Ab der Jahrgangsstufe 9 kann das Fach im Wahlpflichtbereich II (WP II) dreistündig gewählt werden und wird von etwa einem Drittel der Schülerinnen und Schüler besucht. In der zweijährigen Laufzeit dieser Differenzierungskurse wird in altersstufengerechter Weise unter anderem auf Grundlagen der Algorithmik am Beispiel einer didaktischen Lernumgebung, auf die technische Informatik am Beispiel von Schaltwerken und Schaltnetzen eingegangen.

Um insbesondere Schülerinnen und Schülern gerecht zu werden, die in der Sekundarstufe I keinen Informatikunterricht im Wahlpflichtbereich besucht haben, wird in Kursen der Einführungsphase besonderer Wert daraufgelegt, dass keine Vorkenntnisse aus der Sekundarstufe I zum erfolgreichen Durchlaufen des Kurses erforderlich sind.

Der Unterricht der Sekundarstufe II wird mit Hilfe der Programmiersprache Java durchgeführt. In der Einführungsphase kommt dabei zusätzlich eine didaktische Bibliothek zum Einsatz, welche das Erstellen von grafischen Programmen erleichtert.

Durch projektartiges Vorgehen, offene Aufgaben und Möglichkeiten, Problemlösungen zu verfeinern oder zu optimieren, entspricht der Informatikunterricht der Oberstufe in besonderem Maße den Erziehungszielen, Leistungsbereitschaft zu fördern, ohne zu überfordern.

Die gemeinsame Entwicklung von Materialien und Unterrichtsvorhaben, die Evaluation von Lehr- und Lernprozessen sowie die stetige Überprüfung und eventuelle Modifikation des schulinternen Curriculums durch die Fachkonferenz Informatik stellen einen wichtigen Beitrag zur Qualitätssicherung und -entwicklung des Unterrichts dar.

Der Fachschaft Informatik stehen zwei Computerräume mit insgesamt 50 Computerarbeitsplätzen zur Verfügung. Des Weiteren können 120 Laptops zum Lernen genutzt werden. Alle Arbeitsplätze sind an das schulinterne Rechnernetz angeschlossen, so dass Schülerinnen und Schüler über einen individuell gestaltbaren Zugang zum zentralen Server der Schule von jedem Raum Zugriff auf ihre eigenen Daten, zur Recherche im Internet oder zur Bearbeitung schulischer Aufgaben haben.

Der Unterricht erfolgt im 45-Minuten-Takt.

2 Entscheidungen zum Unterricht

2.1 Unterrichtsvorhaben

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.

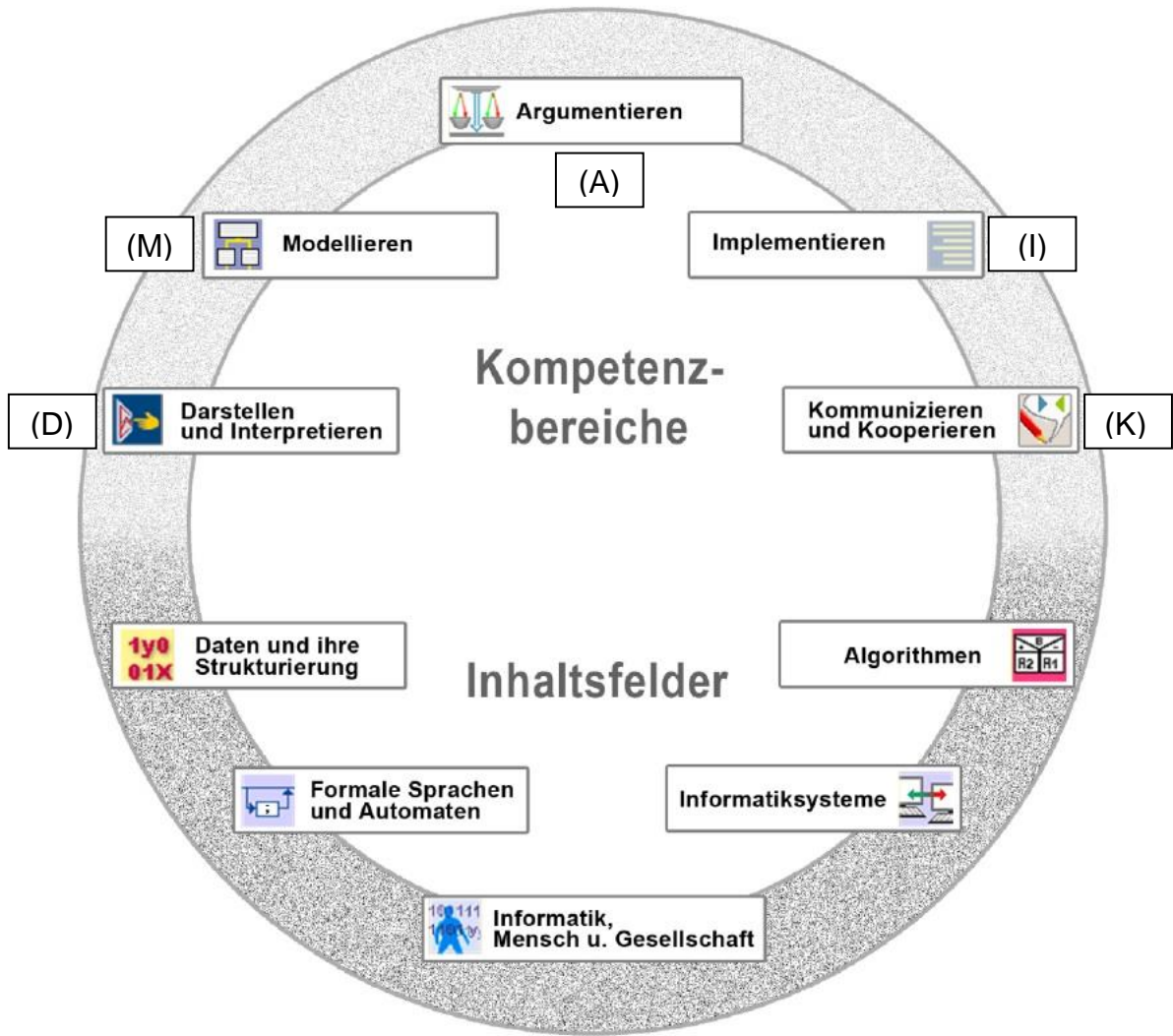
Die Ausweisungen „konkretisierter Unterrichtsvorhaben“ (Kapitel 2.1.1 bis 2.1.3) beinhalten Beispiele und Materialien, die empfehlenden Charakter haben. Referendarinnen und Referendaren sowie neuen Kolleginnen und Kollegen dienen diese vor allem zur standardbezogenen Orientierung in der neuen Schule, aber auch zur Verdeutlichung von unterrichtsbezogenen fachgruppeninternen Absprachen zu didaktisch-methodischen Zugängen, fächerübergreifenden Kooperationen, Lernmitteln und -orten sowie vorgesehenen Leistungsüberprüfungen, die im Einzelnen auch den Kapiteln 2.2 bis 2.3 zu entnehmen sind.

Da in den folgenden Unterrichtsvorhaben Inhalte in der Regel anhand von Problemstellungen in Anwendungskontexten bearbeitet werden, werden in einigen Unterrichtsvorhaben jeweils mehrere Inhaltsfelder angesprochen. Dazu können die folgenden, auf Java basierenden Entwicklungsumgebungen und Programme verwendet werden (in alphabetischer Reihenfolge):

- BlueJ
- Eclipse
- Greenfoot
- Java-Editor
- Java-Kara
- Netbeans

Die oben genannten Entwicklungsumgebungen und Programme können kostenlos aus dem Internet bezogen werden.

In der Ausweisung der konkretisierten Unterrichtsvorhaben werden die zu entwickelnden Kompetenzen und Inhaltsfelder durch einen in Klammern angefügten Großbuchstaben gekennzeichnet, deren Bedeutung der nachfolgenden Grafik zu entnehmen ist. Die Grafik ist aus dem Kernlehrplan für die Sekundarstufe II im Fach Informatik für Gymnasien vom Ministerium für Schule und Weiterbildung des Landes Nordrhein-Westfalen entnommen.



2.1.1 Konkretisierte Unterrichtsvorhaben EF

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: Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten

Leitfragen: *Womit beschäftigt sich die Wissenschaft der Informatik? Wie kann die in der Schule vorhandene informatische Ausstattung genutzt werden?*

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.

Zunächst wird auf den Begriff der Information eingegangen und die Möglichkeit der Kodierung in Form von Daten thematisiert. Anschließend 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.

Des Weiteren 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.

Zeitbedarf: ca. 6 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Materialien
<p>1. Information, deren Kodierung und Speicherung</p> <p>(a) Informatik als Wissenschaft der Verarbeitung von Informationen</p> <p>(b) Darstellung von Informationen in Schrift, Bild und Ton</p> <p>(c) Speichern von Daten mit informatischen Systemen am Beispiel der Schulrechner</p> <p>(d) Vereinbarung von Richtlinien zur Datenspeicherung auf den Schulrechnern (z.B. Ordnerstruktur, Dateibezeichner usw.)</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), • nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (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>2. Informations- und Datenübermittlung in Netzen</p> <p>(a) „Sender-Empfänger-Modell“ und seine Bedeutung für die Eindeutigkeit von Kommunikation</p> <p>(b) Informatische Kommunikation in Rechnernetzen am Beispiel des Schulnetzwerks (z.B. Benutzeranmeldung, Netzwerkordner, Zugriffsrechte, Client-Server)</p> <p>(c) Grundlagen der technischen Umsetzung von Rechnerkommunikation am Beispiel des Internets (z.B. Netzwerkadresse, Paketvermittlung, Protokoll)</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>(d) Richtlinien zum verantwortungsvollen Umgang mit dem Internet</p>		
<p>3. Aufbau informatischer Systeme (a) Identifikation typischer Komponenten informatischer Systeme und anschließende Beschränkung auf das Wesentliche, Identifikation des EVA-Prinzips (Eingabe-Verarbeitung-Ausgabe) als Prinzip der Verarbeitung von Daten</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>

Unterrichtsvorhaben EF-II

Thema: Grundlagen der objektorientierten Analyse, Modellierung und Implementierung

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 oder Beziehungsdigramme eingeführt.

Im Anschluss wird mit der Realisierung erster Projekte mit Hilfe einer didaktischen Programmierumgebung 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.

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: ca. 9 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, 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 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), 	<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>Autos, Lehrerparkplatz Schülerinnen und Schüler betrachten die Kfz auf dem Lehrerparkplatz als Mengen prinzipiell gleichartiger Objekte, die in Klassen 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</p>	<ul style="list-style-type: none"> • stellen den Zustand eines Objekts dar (D). 	
<p>3. Implementierung statischer Szenen</p> <p>(a) Grundaufbau einer Java-Klasse</p> <p>(b) Deklaration und Initialisierung von Objekten</p>		<p><i>Beispiel:</i> Landschaftsszene Schülerinnen und Schüler erstellen ein Programm, das mit Hilfe von geometrischen Objekten eine Landschaft auf den Bildschirm bringt.</p>

(c) Methodenaufrufe mit Parameterübergabe zur Manipulation von Objekteigenschaften (z.B. Farbe, Position)		<i>Beispiel:</i> Olympische Ringe Die Schülerinnen und Schüler bilden das Emblem der olympischen Spiele mit Hilfe von nach.
--	--	--

Unterrichtsvorhaben EF-III

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

Leitfragen: *Wie lassen sich Animationen und Simulationen optischer Gegenstandsbereiche unter Berücksichtigung von Eingaben 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.

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. Auch dieses Projekt soll eine Animation, ggf. im Sinne einer Simulation, sein, bei der Attributwerte von Objekten eigener Klassen verändert werden und diese Veränderungen optisch sichtbar gemacht werden.

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

Zeitbedarf: ca. 18 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, 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) Mehrstufige Animationen mit mehreren sequenziellen Schleifen</p> <p>(c) Berechnung von Abständen zwischen Objekten mit Hilfsvariablen</p> <p>(d) Meldungen zur Kollision zweier Objekte mit Hilfe von Abstandsberechnungen und Verzweigungen (Bedingte- und Alternativanweisungen)</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), • modifizieren einfache Algorithmen und Programme (I), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I), • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), • testen Programme schrittweise anhand von Beispielen (I), 	<p><i>Beispiel: Bewegung</i> Die Schülerinnen und Schüler realisieren mit Objekten Bewegungen auf dem Bildschirm</p>
<p>2. Modellierung und Animation komplexerer grafisch repräsentierbarer Objekte</p> <p>(a) Modellierung eines Simulationsprogramms mit eigenen Klassen mit Hilfe eines Implementationsdiagramms</p> <p>(b) Implementierung eigener Methoden mit und ohne Parameterübergabe</p> <p>(c) Realisierung von Zustandsvariablen</p> <p>(d) Thematisierung des Geheimnisprinzips und des Autonomitätsprinzips von Objekten</p> <p>(e) Animation mit Hilfe des Aufrufs von selbst implementierten Methoden</p> <p>(f) Vertiefung: Weitere Projekte</p>	<ul style="list-style-type: none"> • ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M), • modifizieren einfache Algorithmen und Programme (I), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I), • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), • testen Programme schrittweise anhand von Beispielen (I), 	<p><i>Beispiel: Kerzensimulation</i> Die Schülerinnen und Schüler modellieren und erstellen eine Klasse, mit deren Hilfe Kerzen simuliert werden können. Eine Kerze kann angezündet und gelöscht werden. Abgesehen davon brennen Kerzen abhängig von ihrer Dicke unterschiedlich schnell ab.</p> <p><i>Beispiel: Uhren</i> Die Schülerinnen und Schüler erstellen eine Simulation mehrerer Uhren für verschiedene Zeitzonen.</p> <p><i>Beispiel: Ampeln</i> Die Schülerinnen und Schüler erstellen eine Ampelkreuzung mit mehreren Ampelanlagen an einem Bahnübergang.</p>

	<ul style="list-style-type: none">• interpretieren Fehlermeldungen und korrigieren den Quellcode (I).	
--	---	--

Unterrichtsvorhaben EF-IV

Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen

Leitfrage: Wie werden realistische Systeme anforderungsspezifisch reduziert, als Entwurf modelliert und implementiert? Wie kommunizieren Objekte und wie wird dieses dargestellt und realisiert?

Vorhabenbezogene Konkretisierung:

Das Unterrichtsvorhaben hat die Entwicklung von Objekt -und Klassenbeziehungen zum Schwerpunkt. Dazu werden, ausgehend von der Realität, über Objektidentifizierung und Entwurf bis hin zur Implementation kleine Softwareprodukte in Teilen oder ganzheitlich erstellt.

Zuerst identifizieren die Schülerinnen und Schüler Objekte und stellen diese dar. Aus diesen Objekten werden Klassen und ihre Beziehungen in Entwurfsdiagrammen erstellt.

Nach diesem ersten Modellierungsschritt werden über Klassendokumentationen und der Darstellung von Objektkommunikationen anhand von Sequenzdiagrammen Implementationsdiagramme entwickelt. Danach werden die Implementationsdiagramme unter Berücksichtigung der Klassendokumentationen in Java-Klassen programmiert. In einem letzten Schritt wird das Konzept der Vererbung sowie seiner Vorteile erarbeitet.

Zum Schluss dieses Abschnittes sollen die Schülerinnen und Schüler den Unterschied zwischen Methoden, die „nur“ etwas ausführen, und Methoden, die einen Wert zurückgeben, kennenlernen. Außerdem wird das Methodenkonzept um Parameter erweitert.

Zeitbedarf: ca. 18 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Umsetzung von Anforderungen in Entwurfsdiagramme</p> <p>(a) Aus Anforderungsbeschreibungen werden Objekte mit ihren Eigenschaften identifiziert</p> <p>(b) Gleichartige Objekte werden in Klassen (Entwurf) zusammengefasst und um Datentypen und Methoden erweitert</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>2. Implementationsdiagramme als erster Schritt der Programmierung</p> <p>(a) Erweiterung des Entwurfsdiagramms um Konstruktoren und get- und set-Methoden</p> <p>(b) Festlegung von Datentypen in Java, sowie von Rückgaben und Parametern</p> <p>(c) Entwicklung von Klassendokumentationen</p> <p>(d) Erstellung von Sequenzdiagrammen als Vorbereitung Vorbereitung für die Programmierung</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>3. Programmierung anhand der Dokumentation und des Implementations- und Sequenzdiagrammes</p> <p>(a) Klassen werden in Java-Quellcode umgesetzt</p>	<ul style="list-style-type: none"> • testen Programme schrittweise anhand von Beispielen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), 	

<p>(b) Das Geheimnisprinzip wird umgesetzt</p> <p>(c) Einzelne Klassen und das Gesamtsystem werden anhand der Anforderungen und Dokumentationen auf ihre Korrektheit überprüft.</p>	<ul style="list-style-type: none"> • analysieren und erläutern einfache Algorithmen und Programme (A) • modifizieren einfache Algorithmen und Programme (I), • entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M). • stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D) 	
<p>4. Vererbungsbeziehungen</p> <p>(a) Das Grundprinzip der Vererbung wird erarbeitet</p> <p>(b) Die Vorteile der Vererbungsbeziehungen</p> <p>(c) Vererbung wird implementiert</p>		
<p>5. Rückgaben und Parameter</p> <p>(a) Methoden mit Rückgaben</p> <p>(b) Methoden mit Parametern</p>		

Unterrichtsvorhaben EF-V

Thema: Algorithmen zum Suchen und Sortieren auf Feldern (Arrays) 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 Einführung von Feldern als erster Datenstruktur sowie der Erarbeitung grundlegender, einfacher Such- und Sortieralgorithmen.

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 in einem Pseudocode notiert. Die Schülerinnen und Schüler sollen auf diese Weise das *Sortieren durch Vertauschen*, das *Sortieren durch Auswählen* und das *Sortieren durch Einfügen* kennen lernen und diese im Anschluss implementieren.

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

Zeitbedarf: ca. 9 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>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 (Nutzung eines einzigen oder zweier Felder bzw. lediglich eines einzigen zusätzlichen Ablageplatzes oder mehrerer neuer Ablageplätze)</p> <p>(e) Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der</p>		

<p>Rechenzeit und des Speicherplatzbedarfs</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</p> <p>Mehrere Tischtennisbälle sind nummeriert, sortiert und unter Bechern verdeckt. Mit Hilfe der binären Suche kann sehr schnell ein bestimmter Tischtennisball gefunden werden.</p>

Unterrichtsvorhaben EF-VI

Thema: Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes

Leitfrage: *Welche Entwicklung durchlief die moderne Datenverarbeitung und welche Auswirkungen ergeben sich insbesondere hinsichtlich neuer Anforderungen an den Datenschutz daraus?*

Vorhabenbezogene Konkretisierung:

Das folgende Unterrichtsvorhaben stellt den Abschluss der Einführungsphase dar. Schülerinnen und Schüler sollen selbstständig informatische Themenbereiche aus dem Kontext der Geschichte der Datenverarbeitung und insbesondere den daraus sich ergebenden Fragen des Datenschutzes bearbeiten. Diese Themenbereiche werden in Kleingruppen bearbeitet und in Form von Plakatpräsentationen vorgestellt. Schülerinnen und Schüler sollen dabei mit Unterstützung des Lehrenden selbstständige Recherchen zu ihren Themen anstellen und auch eine sinnvolle Eingrenzung ihres Themas vornehmen.

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: ca. 15 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“ <p>(b) Vorstellung und Diskussion durch Schülerinnen und Schüler</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A), • erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A), • stellen ganze Zahlen und Zeichen in Binärcodes dar (D), • interpretieren Binärcodes als Zahlen und Zeichen (D), • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation. (K). 	<p><i>Beispiel:</i> Ausstellung zu informatischen Themen Die Schülerinnen und Schüler bereiten eine Ausstellung zu informatischen Themen vor. Dazu werden Stellwände und Plakate vorbereitet, die ggf. auch außerhalb des Informatikunterrichts in der Schule ausgestellt werden können.</p> <p><i>Materialien:</i> Schülerinnen und Schüler recherchieren selbstständig im Internet, in der Schulbibliothek, in öffentlichen Bibliotheken, usw.</p>
<p>2. Vertiefung des Themas Datenschutz</p> <p>(a) Erarbeitung grundlegender Begriffe des Datenschutzes</p> <p>(b) Problematisierung und Anknüpfung an die Lebenswelt der Schülerinnen und Schüler</p> <p>(c) Diskussion und Bewertung von Fallbeispielen aus dem Themenbereich „Datenschutz“</p>		<p><i>Beispiel:</i> Fallbeispiele aus dem aktuellen Tagesgeschehen Die Schülerinnen und Schüler bearbeiten Fallbeispiele aus ihrer eigenen Erfahrungswelt oder der aktuellen Medienberichterstattung.</p> <p><i>Materialien:</i> Materialblatt zum Bundesdatenschutzgesetz</p>

2.1.2 Konkretisierte Unterrichtsvorhaben Q1

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 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).

Zusätzliche Kompetenzen für Leistungskurse sind grün hervorgehoben.

Übersichtsraster Unterrichtsvorhaben in der Qualifikationsphase - Q1

Qualifikationsphase – Q1	
<p><u>Unterrichtsvorhaben Q1-I</u></p> <p>Thema: Wiederholung und Vertiefung der objektorientierten Modellierung</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Modellieren - Darstellen und Interpretieren - Implementieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Daten und ihre Strukturierung - Algorithmen - Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Objekte und Klassen - Wirkung der Automatisierung <p>Zeitbedarf: ca. 15 Stunden</p>	<p><u>Unterrichtsvorhaben Q1-II</u></p> <p>Thema: Organisation und Verarbeitung von Daten I – Modellierung und Implementierung von Anwendungen mit dynamischen und linearen Datenstrukturen</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Modellieren - Implementieren - Darstellen und Interpretieren - Argumentieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Daten und ihre Strukturierung - Algorithmen <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Objekte und Klassen - Syntax und Semantik einer Programmiersprache - Analyse, Entwurf und Implementierung von Algorithmen - Algorithmen in ausgewählten informatischen Kontexten <p>Zeitbedarf: ca. 20 Stunden</p>
<p><u>Unterrichtsvorhaben Q1-III</u></p> <p>Thema: Algorithmen zum Suchen und Sortieren auf linearen Datenstrukturen</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Darstellen und Interpretieren - Modellieren - Implementieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Algorithmen - Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Analyse, Entwurf und Implementierung von Algorithmen - Algorithmen in ausgewählten informatischen Kontexten - Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: ca. 20 Stunden</p>	<p><u>Unterrichtsvorhaben Q1-IV</u></p> <p>Thema: Automaten und formale Sprachen</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Darstellen und Interpretieren - Modellieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Formale Sprachen und Automaten - Informatiksysteme <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Syntax und Semantik einer Programmiersprache - Endliche Automaten - Grammatiken regulärer Sprachen - Möglichkeiten und Grenzen von Automaten und formalen Sprachen <p>Zeitbedarf: ca. 20 Stunden</p>
<p>Summe Qualifikationsphase 1: 75 Stunden</p>	

Unterrichtsvorhaben Q1-I

Thema: Wiederholung und Vertiefung der objektorientierten Modellierung

Leitfragen: Wie wird aus einem anwendungsbezogenen Sachkontext ein informatisches Klassenmodell entwickelt? Wie werden Attribute, Methoden und Beziehungen identifiziert, den Klassen zugeordnet und dargestellt? Welche Auswirkungen hat die informatisch-technische Entwicklung auf das Leben der Menschen?

Vorhabenbezogene Konkretisierung:

Der bereits bekannte objektorientierte Zugang zu informatischer Modellierung wird von einer allgemeinen Betrachtung dieses informatischen Konzepts auf eine konkrete Problematik übertragen. Anhand dieser wird eine anwendungsbezogene Implementation Schritt für Schritt von der Objektidentifikation über das Entwurfs- und Implementationsdiagramm durchlaufen.

Grundlegende Modellierungskonzepte wie Sichtbarkeiten, Assoziationen, Vererbung sowie deren Darstellung in Entwurfs- und Klassendiagrammen und Dokumentationen werden wiederholt. Ebenso wird erneut die grafische Darstellung von Objektkommunikation thematisiert.

Anhand von Gütekriterien und Eigenschaften von Modellierung entwickeln und bewerten die Schülerinnen und Schüler Klassenentwürfe.

Das Konzept der objektorientierten Modellierung wird um die Idee der abstrakten Klasse sowie um das Subtyping erweitert.

Zeitbedarf: ca. 15 Stunden

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und mögliche Materialien
<p>1. Wiederholung der grundlegenden Konzepte der objektorientierten Programmierung</p> <p>a) Sichtweise der objektorientierten Informatik auf die Welt</p> <p>b) OOP als informatikspezifische Modellierung der Realität</p> <p>c) Schritte der Softwareentwicklung</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> - analysieren und erläutern objektorientierte Modellierungen (A), - modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), - ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), - modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), 	<p>2.1 Die Welt ist voller Objekte ProjektEinstieg: Klassenentwurf – step by step</p>
<p>2. Erweiterung der objektorientierten Programmierung</p> <p>a) Umsetzung einer Anforderung in Entwurfs- und Klassendiagramm</p> <p>b) Objektkommunikation im Sequenzdiagramm</p> <p>c) Klassendokumentation</p> <p>d) Umsetzung von Teilen der Modellierung</p>	<ul style="list-style-type: none"> - nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), - wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I), - stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), - dokumentieren Klassen (D), - stellen die Kommunikation zwischen Objekten grafisch dar (D), 	<p>2.2 Gut geplant – Klassenentwurf 2.3 Vererbungshierarchien nutzen</p>
<p>3. Mensch und Technik</p> <p>a) Verantwortung von Informatikern</p> <p>b) Automatisierung des Alltags durch Informatik</p>	<ul style="list-style-type: none"> - untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A), - untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A). 	<p>Die digitale Welt 001 - Mensch und Technik</p>
<p>4. Übung und Vertiefung der OOM / OOP</p>	<p>Prüfungsvorbereitung</p>	<p>Prüfungsvorbereitung</p>

Unterrichtsvorhaben Q1-II

Thema:

Organisation und Verarbeitung von Daten I – Modellierung und Implementierung von Anwendungen mit dynamischen und linearen Datenstrukturen

Leitfragen:

Wie müssen Daten linear strukturiert werden, um in den gestellten Anwendungsszenarien eine beliebige Anzahl von Objekten verwalten zu können?

Vorhabensbezogene Konkretisierung:

Ausgehend von einigen Alltagsbeispielen werden als Erstes die Anforderungen an eine Datenstruktur erschlossen. Anschließend werden die Möglichkeiten des Arrays untersucht, lineare Daten zu verwalten und über deren Grenzen/Probleme die Vorteile einer dynamischen linearen Struktur am Beispiel der Struktur Queue erarbeitet (Anwendungskontext Warteschlange). Die Klasse *Queue* selbst wird vorgegeben, die Operationen erläutert. Zur Vertiefung der Kenntnisse wird ein weiteres Anwendungsszenario eingeführt (Polizeikontrolle), dessen Lösung modelliert und implementiert wird. Darauf folgt die Erarbeitung der Struktur Stack, die mithilfe eines einfachen Anwendungsszenarios eingeführt (Biber/Palindrom) wird. Auch hier wird die Klasse *Stack* selbst vorgegeben und die Operationen erläutert. Weitere Aufgaben dienen der Vertiefung und Sicherung.

Um die Unterschiede der beiden Prinzipien FIFO und LIFO zu verstehen, werden zur Lösung der Aufgaben sowohl der Stack als auch die Queue benötigt.

Als letzte lineare dynamische Datenstruktur wird die Liste eingeführt. In dieser Sequenz liegt der Fokus auf der Möglichkeit, auf jedes Element zugreifen zu können. Nachdem die umfangreicheren Standardoperationen dieser Datenstruktur in einem einführenden Beispiel (Vokabeltrainer) erarbeitet und in einem weiteren Beispiel vertieft (LED) wurden, werden abschließend in einem Anwendungskontext verschiedene lineare Datenstrukturen angewendet. Die Modellierung erfolgt beim gesamten Vorhaben in Entwurfs- und Implementationsdiagrammen.

Zeitbedarf: ca. 20 Stunden

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und mögliche Materialien
<p>1. Die Datenstruktur Feld</p> <p>a) Erarbeitung der Anforderungen an eine Datenstruktur</p> <p>b) Wiederholung der Datenstruktur Array, Eigenschaften der Datenstruktur, Standardoperationen für ein und zweidimensionale Arrays</p> <p>c) Modellierung und Implementierung von Anwendungen</p>	<p>Die Schülerinnen und Schüler...</p> <ul style="list-style-type: none"> - erläutern und begründen methodische Vorgehensweisen, Entwurfs- und Implementationsentscheidungen sowie Aussagen über Informatiksysteme (A) - konstruieren zu kontextbezogenen Problemstellungen informatische Modelle (M) - 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), 	<p>Kapitel 3.1:</p> <ul style="list-style-type: none"> - Anforderungen an eine Datenstruktur - Speichern mit Struktur <p><i>neue Wdh. Aufgabe entwickeln, z. B. eine Chart-Top-10, eine Aufgabe mit zweidimensionalem Array (vgl. Anforderungen KLP)</i></p>
<p>2. Die Datenstruktur Schlange</p> <p>a) Modellierung und Implementierung der Verknüpfung von Objekten</p> <p>b) Generische Typen, Trennung von Verwaltung und Inhalt dyn. DS.</p> <p>c) Erläuterung von Problemstellungen, die nach dem FIFO-Prinzip bearbeitet werden</p> <p>d) Funktionalität der Schlange unter Verwendung der Klasse <i>Queue</i>; Erschließen der Standardoperationen</p> <p>e) Modellierung und Implementierung einer Anwendung auf der Basis einer Anforderungsbeschreibung mit Objekten der Klasse <i>Queue</i></p>	<ul style="list-style-type: none"> - implementieren auf der Grundlage von Modellen oder Modellausschnitten Computerprogramme (I) - testen und korrigieren Computerprogramme (I) - interpretieren Fehlermeldungen und korrigieren den Quellcode (I), - überführen gegebene textuelle und grafische Darstellungen informatischer Zusammenhänge in die jeweils andere Darstellungsform (D) - stellen informatische Modelle und Abläufe in Texten, Tabellen, Diagrammen und Grafiken dar (D) - stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D) - modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M) - ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M) - dokumentieren Klassen (D) - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I) - implementieren Operationen dynamischer (linearer und nichtlinearer) Datenstrukturen (A). 	<p>Kapitel 3.2:</p> <ul style="list-style-type: none"> - Wer zuerst kommt - Objekte miteinander verketten - Verwaltung und Inhalt - Funktionen der Queue <p>Mögliche Kontexte:</p> <ul style="list-style-type: none"> - <i>Warteschlange Büro (Standardoperationen/Basiskompetenz) Kunden warten auf einem Flur, um in ein Büro vorgelassen zu werden. Sie können sich am Ende der Warteschlange anstellen, vorgelassen werden oder müssen alle gehen, wenn die Sprechzeit vorüber ist.</i> - <i>Erweiterte Queue Verkehrskontrolle (Vertiefung) Die Polizei kontrolliert die Fahrzeuge im Hinblick auf ihre Verkehrstauglichkeit. Für die Kontrolle werden die Fahrzeuge aus dem Verkehr gewunken. Es werden so lange Fahrzeuge kontrolliert, bis eine gewisse Menge an Verstößen vorliegt oder Autos kontrolliert wurden.</i>
<p>3. Die Datenstruktur Stapel</p> <p>a) Erläuterung von Problemstellungen, die nach dem LIFO-Prinzip bearbeitet werden</p> <p>b) Funktionalität der Klasse Stapel unter Verwendung der Klasse <i>Stack</i></p> <p>c) Modellierung und Implementierung einer Anwendung auf Basis einer Anforderungsbeschreibung mit Objekten der Klasse</p>		<p>Kapitel 3.3:</p> <ul style="list-style-type: none"> - Daten gut abgelegt – Stapel - Funktionen der Datenstruktur Stapel <p>Mögliche Kontexte:</p> <ul style="list-style-type: none"> - <i>Standardoperationen/Basiskompetenz (Stapel Münzen/CDs) zur Umsetzung der gegebenen Funktionen der Klasse Stack</i> - <i>Biber und Teller Es gibt große und kleine Bi-</i>

<p><i>Queue</i></p> <p>d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (Objekte der Klassen <i>Queue</i>, <i>Stack</i> und <i>Array (Palindrom)</i>)</p>		<p><i>ber sowie grüne und braune Teller. Es muss überprüft werden, ob die gestapelten Teller zur Schlange der Biber passen, da die großen Biber nur von den braunen Tellern essen und die kleinen von den grünen. Hierbei müssen sowohl Queue als auch Stack verwendet werden.</i></p> <p><i>Palindrom: Es wird überprüft, ob ein beliebiges Wort ein Palindrom ist.</i></p>
<p>4. Die Datenstruktur Liste</p> <p>a) Analyse der Möglichkeiten bisheriger Datenstrukturen zwecks Bestimmung notwendiger Funktionalitäten für komplexere Anwendungen (Abgrenzung zu <i>Stack/Queue</i>, zusätzliche Fähigkeiten der Klasse <i>List</i>)</p> <p>b) Erarbeitung der Funktionalität der Liste unter Verwendung der Klasse <i>List</i></p> <p>c) Modellierung und Implementierung einer Anwendung mit Objekten der Klasse <i>List</i></p> <p>d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (<i>Stack</i>, <i>Queue</i>, <i>List</i>)</p>		<p>Kapitel 3.4:</p> <ul style="list-style-type: none"> - Flexibel für alle Fälle – (die lineare Liste - Funktionen der Datenstruktur Liste <p>Mögliche Kontexte:</p> <ul style="list-style-type: none"> - LEDs - Textzeilen verarbeiten
<p>5. Übungen und Vertiefungen zur Verwendung linearer und dynamischer Datenstrukturen anhand weiterer Problemstellungen</p>		<p><i>Prüfungsvorbereitung</i></p>
		<p><i>Projekteinstieg Heldenspiel</i></p> <p><i>Mit dem Heldenspiel können alle im Kapitel behandelten Datenstrukturen erarbeitet werden. Das Spiel kann bis zu einem beliebigen Grad realisiert werden, sodass es sowohl als Einstieg als auch als ein umfassendes Projekt für lineare Datenstrukturen genutzt werden kann.</i></p>

Unterrichtsvorhaben Q1-III

Thema: Algorithmen zum Suchen und Sortieren auf linearen Datenstrukturen

Leitfragen: Nach welchen Grundprinzipien können Algorithmen strukturiert werden? Welche Qualitätseigenschaften sollten Algorithmen erfüllen? Wie können mithilfe von Such- und Sortieralgorithmen Daten in linearen Strukturen effizient (wieder-)gefunden werden?

Vorhabenbezogene Konkretisierung:

Zunächst werden anhand eines Anwendungsbeispiels übergreifende Algorithmeigenschaften (wie Korrektheit, Effizienz und Verständlichkeit) erarbeitet und Schritte der Algorithmenentwicklung wiederholt. Dabei kommen Struktogramme zur Darstellung von Algorithmen zum Einsatz.

Als besondere Struktur von Algorithmen wird die Rekursion an Beispielen veranschaulicht und gegenüber der Iteration abgegrenzt. Rekursive Algorithmen werden von den Schülerinnen und Schülern analysiert und selbst entwickelt.

In der zweiten Unterrichtssequenz geht es um die Frage, wie Daten in linearen Strukturen (lineare Liste und Array) (wieder-)gefunden werden können. Die lineare Suche als iteratives und die binäre Suche als rekursives Verfahren werden veranschaulicht und implementiert. Die Bewertung der Algorithmen erfolgt, indem jeweils die Anzahl der Vergleichsoperationen und der Speicherbedarf ermittelt wird.

Möchte man Daten effizient in einer linearen Struktur wiederfinden, so rückt zwangsläufig die Frage nach einer Sortierstrategie in den Fokus. Es wird mindestens ein iteratives und ein rekursives Sortierverfahren erarbeitet und implementiert sowie ihre Effizienz bewertet.

Zeitbedarf: ca. 20 Stunden

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und mögliche Materialien
<p>1. Eigenschaften von Algorithmen</p> <p>a) Qualitätseigenschaften von Algorithmen</p> <p>b) Strukturierung von Algorithmen mit Hilfe der Strategien „Modularisierung“ und „Teile und Herrsche“</p> <p>c) Analyse und Entwicklung von rekursiven Algorithmen</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> - analysieren und erläutern Algorithmen und Programme (A), - modifizieren Algorithmen und Programme (I), - stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D), - entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ und „Backtracking“ (M), 	<p>Kapitel 4.1:</p> <ul style="list-style-type: none"> - Ohne Algorithmen läuft nichts <p>Kapitel 4.2:</p> <ul style="list-style-type: none"> - Teile die Arbeit – rekursive Algorithmen
<p>2. Suchen in Listen und Arrays</p> <p>a) Lineare Suche in Listen und Arrays</p> <p>b) Binäre Suche in einem Array</p> <p>c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf</p>	<ul style="list-style-type: none"> - implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), - testen Programme systematisch anhand von Beispielen (I). - implementieren und erläutern iterative und rekursive Such- und Sortierverfahren unterschiedlicher Komplexitätsklassen (Speicherbedarf und Laufzeitverhalten) (I), - beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A), 	<p>Kapitel 4.3:</p> <ul style="list-style-type: none"> - Suchen – iterativ und rekursiv
<p>3. Sortieren auf Listen und Arrays</p> <p>a) Entwicklung und Implementierung eines iterativen Sortierverfahrens für eine Liste</p> <p>b) Entwicklung und Implementierung eines rekursiven Sortierverfahrens für ein Array</p> <p>c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf</p>	<ul style="list-style-type: none"> - beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), - 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 und mithilfe von Testanwendungen (I), - erläutern das Prinzip der Nebenläufigkeit (A). 	<p>Kapitel 4.4:</p> <ul style="list-style-type: none"> - Sortieren – iterativ und rekursiv

Unterrichtsvorhaben Q-I IV

Thema: Automaten und formale Sprachen

Leitfragen:

Wie lassen sich reale Automaten durch ein Modell formal beschreiben? Wie kann die Art und Weise, wie ein Computer Zeichen (Eingaben) verarbeitet, durch Automaten dargestellt werden? Welche Eigenschaften besitzen Automaten und was können sie leisten? Wie werden sie dargestellt? Wie werden reguläre Sprachen durch eine Grammatik beschrieben? In welchem Verhältnis stehen endliche Automaten und Grammatiken? Welche Anwendungsfälle können durch endliche Automaten und Grammatiken regulärer Sprachen beschrieben werden und welche nicht?

Vorhabensbezogene Konkretisierung:

Ausgehend von der Beschreibung und Untersuchung realer Automaten wird das formale Modell eines endlichen Automaten entwickelt. Neben dem Mealy-Automaten geht es vor allem um den erkennenden endlichen Automaten. Auf die Erarbeitung der Beschreibung folgt die Modellierung eigener Automaten und die Untersuchung bestehender, um die Eigenschaften und Grenzen eines endlichen Automaten zu erkennen. Hierbei wird dessen Verhalten auf bestimmte Eingaben analysiert.

An den Themenkomplex *Endliche Automaten* schließt sich die Erarbeitung von Grammatiken regulärer Sprachen an. Die Untersuchung beginnt bei der Erschließung der formalen Beschreibung und wird mit der Entwicklung von Grammatiken zu regulären Sprachen fortgeführt. Hierbei wird auch die Beziehung von Grammatiken regulärer Sprachen zu endlichen Automaten an Beispielen erarbeitet und analysiert. Hierzu gehört auch die Untersuchung, welche Problemstellungen durch endliche Automaten und reguläre Grammatiken beschrieben werden können und welche nicht.

Zeitbedarf: ca. 20 Std.

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und mögliche Materialien
<p>1. Endliche Automaten</p> <p>a) Erarbeitung der formalen Beschreibung eines Mealy-Automaten und der Darstellungsformen</p> <p>b) Erarbeitung der formalen Beschreibung eines deterministischen endlichen Automaten (DEA) sowie dessen Darstellungsformen; Erschließung der Fachbegriffe Alphabet, Wort, (akzeptierte) Sprache, Determinismus</p> <p>c) Analyse der Eigenschaften von DEA durch die Modellierung eines Automaten zu einer gegebenen Problemstellung, der Modifikation eines Automaten sowie die Überführung der gegebenen Darstellungsform in eine andere</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> - stellen informatische Modelle und Abläufe in Texten, Tabellen, Diagrammen und Grafiken dar (D) - analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens bei bestimmten Eingaben (A) - ermitteln die Sprache, die ein endlicher Automat oder ein Kellerautomat akzeptiert (D) - entwickeln und modifizieren zu einer Problemstellung endlicher Automaten oder Kellerautomaten (M) - stellen endliche Automaten in Tabellen und Graphen dar und überführen sie in die jeweils andere Darstellungsform (D) - entwickeln zur Grammatik einer regulären oder kontextfreien Sprache einen zugehörigen endlichen Automaten oder einen Kellerautomaten (M) - analysieren und erläutern Grammatiken regulärer und kontextfreier Sprachen (A) 	<p>Kapitel 5.1:</p> <ul style="list-style-type: none"> - Vom realen Automaten zum Modell <p><i>Projekteinstieg: Schatzsuche</i></p> <ul style="list-style-type: none"> - Der Mealy-Automat - Der erkennende endliche Automat - Wort und Sprache
<p>2. Grammatiken regulärer Sprachen</p> <p>a) Erarbeitung der formalen Beschreibung einer regulären Grammatik (Sprache, Terminal und Nicht-Terminal, Produktionen und Produktionsvorschriften)</p> <p>b) Analyse der Eigenschaften einer regulären Grammatik durch deren Entwicklung und Modellierung zu einer gegebenen Problemstellung.</p>	<ul style="list-style-type: none"> - modifizieren Grammatiken regulärer und kontextfreier Sprachen (M) - entwickeln zu einer regulären oder kontextfreien Sprache eine Grammatik, die die Sprache erzeugt (M) - entwickeln zur akzeptierten Sprache eines Automaten eine zugehörige Grammatik (M) - beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D),m - zeigen die Grenzen endlicher Automaten und regulärer Grammatiken auf, - modellieren und implementieren Scanner, Parser und Interpreter zu einer gegebenen regulären Sprache (I). 	<p>Kapitel 5.2:</p> <ul style="list-style-type: none"> - Grammatiken regulärer Sprachen
<p>3. Übungen und Vertiefungen</p> <p>Verwendung endlicher Automaten und Grammatiken regulärer Sprachen</p>		<p><i>Prüfungsvorbereitung</i></p>
		<p><i>Projekteinstieg</i></p> <p><i>Erarbeitung der formalen Beschreibung und Überprüfung des Verhaltens eines erkennenden Automaten auf bestimmte Eingaben</i></p>

2.1.3 Konkretisierte Unterrichtsvorhaben Q2

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 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).

Übersichtsraster Unterrichtsvorhaben in der Qualifikationsphase – Q2

Qualifikationsphase – Q2	
<p><u>Unterrichtsvorhaben Q2-I</u></p> <p>Thema: Organisation und Verarbeitung von Daten II – Modellierung und Implementierung von Anwendungen mit dynamischen nicht-linearen Datenstrukturen</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Darstellen und Interpretieren - Modellieren - Implementieren - 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: ca. 23 Stunden</p>	<p><u>Unterrichtsvorhaben Q2-II</u></p> <p>Thema: Nutzung und Modellierung von relationalen Datenbanken in Anwendungskontexten</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 - Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Datenbanken - Algorithmen in ausgewählten informatischen Kontexten - Syntax und Semantik einer Programmiersprache - Sicherheit <p>Zeitbedarf: ca. 20 Stunden</p>
<p><u>Unterrichtsvorhaben Q2-III</u></p> <p>Thema: Aufbau von und Kommunikation in Netzwerken</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 und Rechnernetzwerke - Sicherheit - Nutzung von Informatiksystemen, Wirkungen der Automatisierung <p>Zeitbedarf: ca. 9 Stunden</p>	<p><u>Unterrichtsvorhaben Q2-IV</u></p> <p>Thema: Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Informatiksysteme - Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Von-Neumann-Architektur und die Ausführung maschinennaher Programme - Grenzen der Automatisierbarkeit <p>Zeitbedarf: ca. 6 Stunden</p>
<p>Summe Qualifikationsphase 2: ca. 68 Stunden</p>	

Unterrichtsvorhaben Q2-I

Thema: Organisation und Verarbeitung von Daten II – Modellierung und Implementierung von Anwendungen mit dynamischen nicht-linearen Datenstrukturen

Leitfragen: Wie können Daten mithilfe von Baumstrukturen verwaltet werden? Wie können mit binären Suchbäumen Inhalte sortiert verwaltet werden und welche Vor- und Nachteile bietet dies?

Vorhabenbezogene Konkretisierung:

Anhand des Anwendungskontextes Spielbäume werden zunächst der generelle Aufbau von Baumstrukturen (auch nicht-binäre) und wichtige Grundbegriffe erarbeitet. Die Darstellung von Bäumen mit Knoten und Kanten wird eingeführt.

Anschließend rückt der Fokus auf die binären Bäume, deren rekursiver Aufbau für die Traversierung der Datenstruktur genutzt wird. Die Preorder-Traversierung wird verwendet, um einen gespeicherten Inhalt in einem Binärbaum zu finden (Tiefensuche).

Der Anwendungskontext Ahnenbaum wird mithilfe der Klasse BinaryTree (der Materialien für das Zentralabitur in NRW) modelliert und (ggf. in Teilen) implementiert. Dabei wird u. a. die Erzeugung eines Binärbaums mithilfe der beiden Konstruktoren der Klasse BinaryTree thematisiert.

Möchte man Daten geordnet speichern, so bietet sich die Struktur des binären Suchbaums an. An Beispielen wird zunächst das Prinzip des binären Suchbaums erarbeitet. Die Operationen des Suchens, Einfügens, Löschens und der sortierten Ausgabe werden thematisiert.

Um Daten in einem Anwendungskontext mithilfe eines binären Suchbaums verwalten zu können, müssen sie in eine Ordnung gebracht werden können, d. h. sie müssen vergleichbar sein. Diese Vorgabe wird mithilfe des Interfaces Item realisiert, das alle Klassen, dessen Objekte in einem Suchbaum verwaltet werden sollen, implementieren müssen. Auf diese Weise wird ein Anwendungskontext (Benutzerverwaltung) mithilfe der Klassen BinarySearchTree und Item modelliert und (ggf. in Teilen) implementiert.

Zeitbedarf: ca. 23 Stunden

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Aufbau von Baumstrukturen und Grundbegriffe</p> <p>a) Erarbeitung der Begriffe Wurzel, Knoten, Blatt, Kante, Grad eines Knotens und eines Baumes, Pfad, Tiefe, Ebene, Teilbaum</p> <p>b) Aufbau und Darstellung von Baumstrukturen in verschiedenen Anwendungskontexten</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> - stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D), - erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), - analysieren und erläutern Algorithmen und Programme (A), - stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D). - beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), 	<p>6.1 Spielen mit Struktur – Baumstrukturen</p> <p>Projekteinstieg 1: Spielbäume</p>
<p>2. Binäre Bäume</p> <p>a) rekursiver Aufbau eines binären Baums</p> <p>b) Traversierungen (pre-, in-, postorder)</p> <p>c) Modellierung eines Binärbaums in einem Anwendungskontext mit Hilfe der Klasse BinaryTree (als Entwurfs- und Implementationsdiagramm)</p> <p>d) Implementation einer Anwendung der Datenstruktur binärer Baum (ggf. in Teilen)</p>	<ul style="list-style-type: none"> - ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), - ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), - modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), - verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M), - entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M), 	<p>6.2 Zwei Nachfolger sind genug! - Binäre Bäume</p> <p>Implementation des Projekts Ahnenbaum</p>
<p>3. Binäre Suchbäume</p> <p>a) Prinzip des binären Suchbaums, Ordnungsrelation</p> <p>b) Operationen auf dem binären Suchbaum (Suchen, Einfügen, Löschen, sortierte Ausgabe)</p> <p>c) Modellierung eines binären Suchbaums in einem Anwendungskontext mit Hilfe der Klasse BinarySearchTree (als Entwurfs- und Implementationsdiagramm) und dem Interface Item</p> <p>d) Implementation einer Anwendung der Datenstruktur binärer Suchbaum (ggf. in Teilen)</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), 	<p>6.3 Wer Ordnung hält, spart Zeit beim Suchen – Binäre Suchbäume</p> <p>Projekteinstieg 2: Binäre Suchbäume</p> <p>Implementation des Projekts Benutzerverwaltung</p>
<p>4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen</p>		<p>Prüfungsvorbereitung</p>

Unterrichtsvorhaben Q2-II

Thema: Nutzung und Modellierung von relationalen Datenbanken in Anwendungskontexten

Leitfragen: Was sind Datenbanken und wie kann man mit ihnen arbeiten? Wie entwickelt man selbst eine Datenbank für einen Anwendungskontext?

Vorhabenbezogene Konkretisierung:

Am Beispiel eines Online-Buchhandels wird der Aufbau einer Datenbank sowie wichtige Grundbegriffe erarbeitet. Die Schülerinnen und Schüler nehmen dabei zunächst die Sicht der Anwender an, die eine bestehende Datenbank beschreiben und analysieren und mithilfe von SQL-Abfragen Daten gezielt herausfiltern.

Mithilfe des Projekteinstiegs „Tabellen“ können bereits zu einem frühen Zeitpunkt des Unterrichtsvorhabens Redundanzen, Inkonsistenzen und Anomalien problematisiert werden.

Nachdem die Lernenden in der ersten Sequenz mit Datenbanken vertraut gemacht wurden, nehmen sie nun die Rolle der Entwickler an, indem sie selbst Datenbanken von Grund auf modellieren und das Modell in ein Relationenschema überführen. Sie arbeiten mit Entity-Relationship-Diagrammen, um Entitäten, zugehörige Attribute, Relationen und Kardinalitäten in Anwendungskontexten darzustellen. Gegebene ER-Diagramme werden analysiert, erläutert und modifiziert. Der bereits in der ersten Sequenz problematisierte Begriff der Redundanz wird am Ende des Unterrichtsvorhabens wieder aufgegriffen, um die Normalisierung von Datenbanken zu thematisieren. Bestehende Datenbankschemata werden hinsichtlich der 1. bis 3. Normalform untersucht und (soweit nötig) normalisiert.

Zeitbedarf: ca. 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Nutzung von relationalen Datenbanken a) Aufbau von Datenbanksystemen und Grundbegriffe - Aufgaben und Eigenschaften eines Datenbanksystems - Erarbeitung der Begriffe Tabelle, Attribut, Attributwert, Datensatz, Datentyp, Primärschlüssel, Datenbankschema - Problematisierung von Redundanzen, Anomalien und Inkonsistenzen b) SQL-Abfragen - Erarbeitung der grundlegenden Sprachelemente von SQL	Die Schülerinnen und Schüler - erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A), - analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A), - analysieren und erläutern eine Datenbankmodellierung (A), - erläutern die Eigenschaften normalisierter Datenbankschemata (A), - bestimmen Primär- und Sekundärschlüssel (M), - ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M), - modifizieren eine Datenbankmodellierung (M),	Kapitel 8 Datenbanken 8.1 Wissen speichern und verwalten – Datenbanksysteme 8.2 Daten anordnen mit Tabellen Beispiel: Buchhandlung Redundanzen, Anomalien und Inkonsistenzen Projekteinstieg: Tabellen 8.3 Daten filtern mit SQL 8.4 Komplexe Filter Aufgaben

<p>(SELECT(DISTINCT), FROM, WHERE, JOIN)</p> <p>- Analyse und Erarbeitung von SQL-Abfragen (AND, OR, NOT, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, <>, >, <, >=, <=, LIKE, BETWEEN, IN, IS NULL, geschachtelte Select-Ausdrücke)</p> <p>c) Vertiefung an einem weiteren Datenbankbeispiel</p>	<ul style="list-style-type: none"> - modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M), - überführen Datenbankschemata in vorgegebene Normalformen (M), - verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I), - ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D), - stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D), 	
<p>2. Modellierung von relationalen Datenbanken</p> <p>a) Datenbankentwurf durch ER-Diagramme</p> <ul style="list-style-type: none"> - Ermittlung von Entitäten, zugehörigen Attributen, Beziehungen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms - Erläuterung und Erweiterung einer Datenbankmodellierung <p>b) Entwicklung eines relationalen Modells aus einem Datenbankentwurf</p> <ul style="list-style-type: none"> - Überführung eines Entity-Relationship-Diagramms in ein relationales Datenbankschema inklusive der Bestimmung von Primär- und Fremdschlüsseln <p>c) Normalformen</p> <ul style="list-style-type: none"> - Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten) 	<ul style="list-style-type: none"> - überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D), - implementieren ein relationales Datenbankschema als Datenbank (I). 	<p>8.5 Datenbankentwurf</p> <p><i>Beispiel: Online-Buchhandel</i></p> <p><i>Datenanalyse und Entwurf</i></p> <p>8.6 Umsetzung des ER-Modells</p> <p><i>Entitätsmengen</i></p> <p><i>m:n-Beziehungen</i></p> <p><i>1:n-Beziehungen</i></p> <p><i>1:1-Beziehungen</i></p> <p><i>Wiederaufgriff des Projekteinstiegs</i></p> <p>8.7 Datenbanken verbessern durch Normalformen</p>
<p>3. Übung und Vertiefung der Nutzung und Modellierung von relationalen Datenbanken</p>		<p>Prüfungsvorbereitung</p>

Unterrichtsvorhaben Q2-III

Thema: Aufbau von und Kommunikation in Netzwerken

Leitfragen: Was macht menschliche Kommunikation aus? Welchen Stellenwert haben technische/ informatische Hilfsmittel für die Kommunikation? Wie werden Daten in einem Netzwerk zwischen den Kommunikationspartnern übertragen? Wie ist die Arbeitsteilung in Netzwerken gestaltet? Wie kann sicher in Netzwerken kommuniziert werden?

Vorhabenbezogene Konkretisierung:

Ausgehend von alltäglicher Face-to-Face-Kommunikation werden die Grundprinzipien sowie die Bewertungskriterien von Kommunikation erläutert. Das Netzwerk wird als vorteilhafte Kommunikationsstruktur dargestellt und anhand von Topologien und Reichweiten kategorisiert. Ausgehend davon wird der Protokollbegriff entwickelt und anhand des TCP/IP-Schichtenmodells analysiert. Anschließend wird das Client-Server-Prinzip vorgestellt und angewandt.

Sichere Kommunikation in Netzen ist nur dank kryptografischer Verfahren möglich. Stellvertretend werden zwei symmetrische und ein asymmetrisches Verfahren erläutert, angewandt und bewertet.

Zeitbedarf: 9 Stunden

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Technische Kommunikation als Fortführung natürlicher Kommunikation</p> <p>a) Kommunikation im Shannon-Weaver-Modell</p> <p>b) Kriterien von technischen Kommunikationsarten</p> <p>c) Die Geschichte der technischen Kommunikation</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), - nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D), - analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A), 	<p>7.1 Menschen kommunizieren – ohne und mit Technik</p> <p>Projekteinstieg: Kommunikation im Wilden Westen</p>
<p>2. Aufbau von Netzwerken und Kommunikationsregeln</p> <p>a) Das Netzwerk als Organisationsprinzip der Kommunikation und Möglichkeiten der Ausformung</p> <p>b) Geregelt technische Kommunikation durch Protokolle in Schichtenmodellen</p>	<ul style="list-style-type: none"> - analysieren und erläutern Algorithmen und Methoden zur Client-Server-Kommunikation (A), - entwickeln und implementieren Algorithmen und Methoden zur Client-Server-Kommunikation (I), - analysieren und erläutern Protokolle zur Kommunikation in einem Client-Server-Netzwerk (A), - entwickeln und erweitern Protokolle zur Kommunikation in einem Client-Server-Netzwerk (M), 	<p>7.2 Ohne Protokoll läuft nichts - Netzwerke</p>
<p>3. Aufgabenteilung in Netzwerken durch Server und Client</p> <p>a) Aufbau und Aufgaben der Client-Server-Struktur</p> <p>(b) Protokolle zwischen Client und Server</p>	<ul style="list-style-type: none"> - nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung von Daten, zur Organisation von Arbeitsabläufen sowie zur Verteilung und Zusammenführung von Arbeitsanteilen (K), 	<p>7.3 Einer für alle – Client-Server-Struktur</p>
<p>4. Kryptologie</p> <p>a) Veranschaulichen und Anwenden von symmetrischen und asymmetrischen kryptographischen Verfahren (Caesar, Vigenère, RSA)</p> <p>b) Bewertung der Verfahren hinsichtlich ihrer Sicherheit und ihrem Aufwand</p>	<ul style="list-style-type: none"> - entwickeln mit didaktische orientierten Entwicklungsumgebungen einfache Benutzungsoberflächen zur Kommunikation mit einem Informatiksystem (M), 	<p>Die digitale Welt 100 – Kryptologie</p>
<p>5. Übung und Vertiefung des Aufbaus von und der Kommunikation in Netzwerken</p>		<p>Prüfungsvorbereitung</p>

Unterrichtsvorhaben Q-II 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?

Vorhabensbezogene 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: ca. 6 Std.

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und mögliche 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> <p>- erläutern die Ausführung eines einfachen maschinennahen Programms sowie die Datenspeicherung auf einer „Von-Neumann-Architektur“ (A),</p> <p>- untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A).</p>	<p>Beispiel:</p> <p>Addition von 4 zu einer eingegebenen Zahl mit einem Rechnermodell</p> <p>Materialien:</p> <p>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 – Von-Neumann-Architektur und maschinennahe Programmierung</p> <p>Schöningh Informatik 1 Exkurs – von-Neumann-Rechner</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>Beispiel: Halteproblem</p> <p>Materialien:</p> <p>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 - Halteproblem</p> <p>(Download Q2-III.2)</p>

2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

In Absprache mit der Lehrerkonferenz sowie unter Berücksichtigung des Schulprogramms hat die Fachkonferenz Informatik die folgenden fachmethodischen und fachdidaktischen Grundsätze beschlossen. In diesem Zusammenhang beziehen sich die Grundsätze 1 bis 14 auf fächerübergreifende Aspekte, die auch Gegenstand der Qualitätsanalyse sind, die Grundsätze 15 bis 21 sind fachspezifisch angelegt.

Überfachliche Grundsätze:

- 1) Geeignete Problemstellungen zeichnen die Ziele des Unterrichts vor und bestimmen die Struktur der Lernprozesse.
- 2) Inhalt und Anforderungsniveau des Unterrichts entsprechen dem Leistungsvermögen der Schüler/innen.
- 3) Die Unterrichtsgestaltung ist auf die Ziele und Inhalte abgestimmt.
- 4) Medien und Arbeitsmittel sind schülernah gewählt.
- 5) Die Schüler/innen erreichen einen Lernzuwachs.
- 6) Der Unterricht fördert eine aktive Teilnahme der Schüler/innen.
- 7) Der Unterricht fördert die Zusammenarbeit zwischen den Schülern/innen und bietet ihnen Möglichkeiten zu eigenen Lösungen.
- 8) Der Unterricht berücksichtigt die individuellen Lernwege der einzelnen Schüler/innen.
- 9) Die Schüler/innen erhalten Gelegenheit zu selbstständiger Arbeit und werden dabei unterstützt.
- 10) Der Unterricht fördert strukturierte und funktionale Partner- bzw. Gruppenarbeit.
- 11) Der Unterricht fördert strukturierte und funktionale Arbeit im Plenum.
- 12) Die Lernumgebung ist vorbereitet; der Ordnungsrahmen wird eingehalten.
- 13) Die Lehr- und Lernzeit wird intensiv für Unterrichtszwecke genutzt.
- 14) Es herrscht ein positives pädagogisches Klima im Unterricht.

Fachliche Grundsätze:

- 15) Der Unterricht unterliegt der Wissenschaftsorientierung und ist dementsprechend eng verzahnt mit seiner Bezugswissenschaft.
- 16) Der Unterricht ist problemorientiert und soll von realen Problemen ausgehen und sich auf solche rückbeziehen.
- 17) Der Unterricht folgt dem Prinzip der Exemplarizität und soll ermöglichen, informatische Strukturen und Gesetzmäßigkeiten in den ausgewählten Problemen und Projekten zu erkennen.
- 18) Der Unterricht ist anschaulich sowie gegenwarts- und zukunftsorientiert und gewinnt dadurch für die Schülerinnen und Schüler an Bedeutsamkeit.
- 19) Der Unterricht ist handlungsorientiert, d.h. projekt- und produktorientiert angelegt.
- 20) Im Unterricht werden sowohl für die Schule didaktisch reduzierte als auch reale Informatiksysteme aus der Wissenschafts-, Berufs- und Lebenswelt eingesetzt.
- 21) Der Unterricht beinhaltet reale Begegnung mit Informatiksystemen.

2.3 Grundsätze der Leistungsbeurteilung

Absprachen zur Leistungsbeurteilung

Der gesamte Bereich der Leistungsbeurteilung soll den Eltern und den Schüler/-innen zu Beginn des Schuljahres mitgeteilt werden, damit Eltern und Schüler/-innen eine klare Orientierung von den zu erreichenden Kenntnissen und Qualifikationen einerseits und deren Überprüfung andererseits gewinnen können und damit eine Basis für eine erfolgreiche Unterrichtsmitarbeit erlangen können.

Leistungsbeurteilung

Die Abschlussnote ergibt sich aus den Noten der schriftlichen Arbeiten und den Leistungen im Beurteilungsbereich „Sonstige Mitarbeit“. Die Bildung der Endnote bestimmen pädagogische Gründe, eine Endnote nur nach dem arithmetischen Mittel ist nicht zulässig.

Schriftlicher Bereich in der Sekundarstufe I und II

Kursarbeiten

Jahrgangsstufe	Anzahl pro Halbjahr	Länge (in 45-Minuten-Stunden)
5	-	-
6	-	-
9	2	2
10	2	2

Klausuren

Jahrgangsstufe	Anzahl pro Halbjahr	Länge (in 45-Minuten-Stunden)
10 (EF)	1	2
11 (Q1)	2	2
12 (Q2)	2	3

Die Fachschaft Informatik orientiert sich bei der Bewertung von Klassenarbeiten in der Sekundarstufe I an folgendem Punktesystem:

sehr gut (100% - 87%), gut (86% - 73%), befriedigend (72% - 59%), ausreichend (58% - 45%), mangelhaft (44% - 20%), ungenügend (19% - 0%).

Bei der Bewertung von Klausuren in der gymnasialen Oberstufe orientiert sich die Fachschaft Informatik an folgendem Punktesystem, das auch dem Zentralabitur zugrunde liegt:

sehr gut (100% - 85%), gut (84% - 70%), befriedigend (69% - 55%), ausreichend (54% - 39%), mangelhaft (38% - 20%), ungenügend (19% - 0%). Die Note „ausreichend minus“ umfasst den Bereich „45% - 39%“.

Mündlicher Bereich / Sonstige Mitarbeit

Hausaufgaben

Die einzelne Hausaufgabe wird in der Regel nicht zensiert. Unter pädagogischen Aspekten sollten Hausaufgaben Anerkennung finden. Hausaufgaben werden regelmäßig überprüft und für die weitere Arbeit im Unterricht ausgewertet.

Mündliche Unterrichtsbeiträge

- Beiträge im Unterrichtsgespräch
- Vortrag bzw. Mitarbeit bei Gruppen- und Partnerarbeiten
- Referate
- Umgang mit der Fachsprache
- Vortrag von selbst erstellten Programmen

Fertigkeiten im Umgang mit dem Medium Computer und Programmiersystemen

Schriftliche Unterrichtsbeiträge

- Kontinuierliche Dokumentation der Unterrichtsinhalte (Heft/Mappe/Ordner)
- Projektbegleitende Berichtsmappen
- Protokolle
- Präsentationsprodukte
- Schriftliche Überprüfung der Hausaufgabe
- Selbst erstellte Programme
- Schriftliche Übungen

Sie sollen angekündigt werden, an den präzisierten Kompetenzen ausgerichtet sein und auf einen überschaubaren Zeitraum (maximal Reihenthema) begrenzt sein. Die zu überprüfenden Themen sollten mit den Schüler/-innen abgesprochen sein bzw. ihnen mitgeteilt werden. Sie sollen auf Reproduktions- und einfache Anwendungsaufgaben begrenzt sein, weiterführende Transferleistungen (Beurteilungen und Kreativlösungen) sollten nicht Gegenstand der Leistungsüberprüfung sein. Die Anzahl der *Schriftlichen Übungen* pro Halbjahr ist an der Wochenstundenzahl des Faches auszurichten, bei zwei Wochenstunden also in der Regel maximal zwei. Die zeitliche Dauer der *Schriftlichen Übungen* soll zwischen 20 und 30 Minuten liegen.

Auf *Schriftliche Übungen* kann verzichtet werden, wenn der Leistungsstand auch über andere Formen der sonstigen Mitarbeit beurteilt werden kann, hierbei entscheidet die Fachkraft in Absprache mit der Fachkonferenz.

Grundlegende Kriterien der Benotung in der Sekundarstufe I und II

Wichtige Kriterien bei der Leistungsbeurteilung sind die Qualität und Kontinuität der Beiträge, die Verwendung der Fachsprache, der richtige Umgang mit den fachlichen Grundbegriffen, also die inhaltliche Leistung und die Darstellungsleistung.

Eine Leistung wird mit **ungenügend** benotet, wenn sie den Anforderungen nicht entspricht, die Grundkenntnisse so lückenhaft sind, dass „die Mängel in absehbarer Zeit nicht behoben werden können.“ (APO GOST §16)

Eine Leistung, die noch notwendige Grundkenntnisse erkennen lässt, aber den Anforderungen nicht entspricht, wird mit **mangelhaft** benotet.

Ausreichend ist eine Leistung, die zumindest auf der Ebene der Reproduktion einfache Fakten und Zusammenhänge im Wesentlichen richtig wiedergibt.

Die Note **befriedigend** entspricht einer Leistung, bei der Zusammenhänge richtig auf der Grundlage entsprechender Kenntnisse hergestellt werden.

Leistungen, die in einer sauber verwendeten Fachsprache einen Zusammenhang gestützt auf die nötigen Fakten differenziert und reflektiert darstellen, werden mit **gut** beurteilt.

Wurde die oben beschriebene Leistung in besonderem Maße erfüllt, wird sie mit **sehr gut** beurteilt.

Schriftliche Arbeiten in der Sekundarstufe II

Die jeweils geltenden Bestimmungen für das Zentralabitur legen die Themenbereiche der Klausuren fest. Ebenso sind die Aufgabenarten festgelegt. Die erste Klausur in der Q1.2 kann durch eine Facharbeit ersetzt werden. Bei der Benotung der Klausuren gilt das auch im Zentralabitur angelegte Punkteschema.

Der Beurteilungsbereich der Sonstigen Mitarbeit in der Sekundarstufe II

Zum Beurteilungsbereich der SoMi „gehören alle im Zusammenhang mit dem Unterricht erbrachten schriftlichen, mündlichen und praktischen Leistungen mit Ausnahme der Klausuren und der Facharbeit gemäß §14 Abs.3“ APO GOST §15 Die Beteiligung am Unterrichtsgespräch, Hausaufgaben, Referate, kurze Vorträge, Zusammenfassungen von Unterrichtsergebnissen, Protokolle, die Präsentation von Ergebnissen, Gruppenarbeit, Arbeitsmappen, Projektbeiträge, praktische Arbeiten, die schriftliche Übung sind Gegenstände des Beurteilungsbereichs SoMi.

Die **Hausaufgabe** hat in der gymnasialen Oberstufe eine wichtige Funktion der Vorbereitung, Vertiefung und Problematisierung. „Hausaufgaben können in der Oberstufe in die Bewertung einbezogen werden, dürfen allerdings nicht als solche im Einzelnen benotet werden, sondern können nur als Gesamteindruck mit in die Bewertung einfließen.“ (APO GOST §15 Erl.)

Referat, Präsentation, praktische Arbeiten u.a. dienen der Wissenschaftspropädeutik. Beurteilungskriterien sind die saubere Verarbeitung der Informationen, die Präzision, die Darstellungs- und Verstehensleistung, die Selbständigkeit, der funktionale Einsatz von Medien, die intentions- und adressatengerechte Präsentation.

Die **schriftliche Übung** ist eine weitere schriftliche Leistung im Rahmen der SoMi - Note. „Die Übung darf sich nur auf begrenzte Stoffbereiche im unmittelbaren Zusammenhang mit dem jeweiligen Unterricht beziehen. ... sie soll maximal die letzten 6 Stunden des Unterrichts abdecken. Die mit „kurz“ beschriebene Zeitdauer der Übung ist mit 20 bis 30 Minuten ausgeschöpft. Die zulässige Zahl der schriftlichen Übungen ist mit „gelegentlich“ zu bezeichnen; die Fachlehrerin und der Fachlehrer haben hier einen gewissen Ermessensspielraum und können berücksichtigen, ob in dem Fach Klausuren geschrieben werden oder nicht, grundsätzlich sollten jedoch die schriftlichen Übungen auf 1 bis 2 Übungen je Fach begrenzt werden.“ (APO GOST §15 Erl. 4)

Die Beurteilung von mündlichen Unterrichtsbeiträgen

Die Schülerinnen und Schüler der gymnasialen Oberstufe sollen von sich aus im Unterricht mitarbeiten.

Eine **nicht ausreichende** Leistung ist dann gegeben, wenn der Schüler / die Schülerin im Unterricht nicht mitarbeitet, die Äußerungen in der Regel fachlich falsch oder nur teilweise richtig sind. „Der Kurslehrer oder die Kurslehrerin muss aber auch die auf wenig Mitarbeit ausgerichteten Schüler in den Lernprozess integrieren. Schülern und Schülerinnen mit begrenzter Leistungsfähigkeit ist planmäßig Gelegenheit zu Beiträgen zum Unterrichtsgespräch zu geben. Bei Notenbeschwerden muss der Lehrer oder die Lehrerin imstande sein, diese kontinuierliche Bemühung nachzuweisen, wobei er frei darin ist, in welcher Form er dies für eine ggf. erforderliche Darstellung festhält.“ (APO GOST §15 Erl.4)

Eine **ausreichende oder befriedigende** Leistung erbringt, wer regelmäßig im Unterricht mitarbeitet. Wenn sich Äußerungen über das Maß der einfachen, aber richtigen Wiedergabe von Fakten zur sprachlich angemessenen Darstellung von Zusammenhängen bewegen, entspricht das den Anforderungen an eine befriedigende Leistung.

Ein Schüler / eine Schülerin, der / die mitarbeitet, auch schwierige Sachverhalte auf der Grundlage von Kenntnissen versteht, Wesentliches vom Unwesentlichen unterscheiden kann und Beiträge zur Problemlösung leisten kann, erbringt eine **gute**, je nach Umfang auch **sehr gute** Leistung.

3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen

Die Fachkonferenz Informatik hat sich im Rahmen des Schulprogramms für folgende zentrale Schwerpunkte entschieden:

Zusammenarbeit mit anderen Fächern

Im Informatikunterricht werden Kompetenzen anhand informatischer Inhalte in verschiedenen Anwendungskontexten erworben, in denen Schülerinnen und Schülern aus anderen Fächern Kenntnisse mitbringen können. Diese können insbesondere bei der Auswahl und Bearbeitung von Softwareprojekten berücksichtigt werden und in einem hinsichtlich der informatischen Problemstellung angemessenem Maß in den Unterricht Eingang finden. Da im Inhaltsfeld Informatik, Mensch und Gesellschaft auch gesellschaftliche und ethische Fragen im Unterricht angesprochen werden, soll eine mögliche Zusammenarbeit mit den Fächern Sozialwissenschaften und Philosophie in einer gemeinsamen Fachkonferenz ausgelotet werden.

Vorbereitung auf die Erstellung der Facharbeit

Möglichst schon im zweiten Halbjahr der Einführungsphase, spätestens jedoch im ersten Halbjahr des ersten Jahres der Qualifikationsphase werden im Unterricht an geeigneten Stellen Hinweise zur Erstellung von Facharbeiten gegeben. Das betrifft u. a. Themenvorschläge, Hinweise zu den Anforderungen und zur Bewertung. Es wird vereinbart, dass nur Facharbeiten vergeben werden, die mit der eigenständigen Entwicklung eines Softwareproduktes verbunden sind.

Exkursionen

In der Einführungsphase wird im Rahmen des Unterrichtsvorhabens „Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes“ eine Exkursion zum Arithmeum in Bonn durchgeführt. Die außerunterrichtliche Veranstaltung wird im Unterricht vor- und nachbereitet.

4 Qualitätssicherung und Evaluation

Durch Diskussion der Aufgabenstellung von Klausuren in Fachdienstbesprechungen und eine regelmäßige Erörterung der Ergebnisse von Leistungsüberprüfungen wird ein hohes Maß an fachlicher Qualitätssicherung erreicht.