

Heinrich-Böll-Gesamtschule



Schulinterner Lehrplan zum Kernlehrplan für die gymnasiale Oberstufe

Informatik

(Stand: 09.03.2016)

Inhaltsverzeichnis

1 Die Fachgruppe Informatik der Heinrich-Böll-Gesamtschule.....	3
2 Entscheidungen zum Unterricht.....	5
2.1 Unterrichtsvorhaben.....	5
2.1.1 Übersichtsraster Unterrichtsvorhaben.....	6
2.1.2 Konkretisierte Unterrichtsvorhaben.....	11
2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit.....	53
2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung.....	55
2.3.1 Beurteilungsbereich Klausuren.....	55
2.3.2 Beurteilungsbereich Sonstige Mitarbeit.....	56
3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen.....	61
4 Qualitätssicherung und Evaluation.....	62

1 Die Fachgruppe Informatik der Heinrich-Böll-Gesamtschule

Bei der Heinrich-Böll-Gesamtschule handelt es sich um eine in der Sekundarstufe I achtzügige und in der Sekundarstufe II fünfzügige Gesamtschule in Köln-Chorweiler. Derzeit besuchen ca. 1650 Schülerinnen und Schüler die Heinrich-Böll-Gesamtschule.

In der Sekundarstufe I bietet die Heinrich-Böll-Gesamtschule Forderkurse in Informatik an. Diese finden als wöchentliche Doppelstunde im Rahmen der Ergänzungsstunden im 9. und 10. Jahrgang statt. Die Kurse richten sich vor allem an leistungsstärkere Schülerinnen und Schüler der Sekundarstufe I.

Neben der Erarbeitung grundlegender Begriffe zum Themenbereich Hardware/Software geht es darum Kompetenzen in der Bedienung klassischer Office-Programme zu vermitteln (Textverarbeitung, Tabellenkalkulation, Datenbanken, Präsentationen). Darüber hinaus sollen die SuS in die Lage versetzt werden, eigene Web-Seiten mittels HTML zu erstellen.

Ein weiteres Themenfeld ist das Sammeln erster Erfahrungen im Bereich des objektorientierten Programmierens. Hierbei starten die SuS zunächst mit Hilfe einer grafischen Programmieroberfläche, wie man sie bei Kara oder LEGO MINDSTORMS findet. Im Anschluss sollen die SuS erste Kenntnisse im Umgang mit der objektorientierten Programmiersprache Java-Script erlangen.

In der Sekundarstufe II bietet die Heinrich-Böll-Gesamtschule in allen Jahrgangsstufen jeweils einen Grundkurs in Informatik an.

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

Der Unterricht der Sekundarstufe II wird mit der Programmiersprache Java durchgeführt. In der Einführungsphase kommen dabei die Entwicklungsumgebungen BlueJ und Greenfoot zum Einsatz.

Durch projektartiges Vorgehen, offene Aufgaben und Möglichkeiten, Problemlösungen zu finden, zu verfeinern oder zu optimieren, entspricht der Informatikunterricht der Oberstufe in besonderem Maße dem Erziehungsziel, 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.

Zurzeit verfügt die Heinrich-Böll-Gesamtschule über 3 Computerräume mit je 21 Arbeitsplätzen, 2 Computerräume mit je 33 Arbeitsplätzen und ein Oberstufen-Selbstlernzentrum mit 15 Arbeitsplätzen. Alle Klassen- und Fachräume sind mit einer Medienecke ausgestattet. Alle Arbeitsplätze sind an das schulinterne Rechnernetz angeschlossen, so dass Schülerinnen und Schüler alle

Arbeitsplätze der Schule zum Zugriff auf ihre eigenen Daten, zur Recherche im Internet oder zur Bearbeitung schulischer Aufgaben verwenden können.

Die Fachschaft Informatik besteht aus 4 Lehrkräften, denen die 5 Computerräume für ihren Unterricht zur Verfügung stehen. 2 Lehrkräfte unterrichten ausschließlich in der Sekundarstufe I, die beiden anderen ausschließlich in der Sekundarstufe II.

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 Kompetenzwertwartungen des Kernlehrplans von ihnen erfüllt werden können.

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

Im „Übersichtsraster Unterrichtsvorhaben“ (Kapitel 2.1.1) 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 zu ermöglichen, bzw. die Erfordernisse anderer besonderer Ereignisse (z.B. Praktika, Kursfahrten o.ä.) zu berücksichtigen, 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 „konkretisierte Unterrichtsvorhaben“ (Kapitel 2.1.2) Beispiele und Materialien, die empfehlenden Charakter haben. Für Referendarinnen und Referendare 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.

2.1.1 Übersichtsraster Unterrichtsvorhaben

I) Einführungsphase

Einführungsphase	
<p><u>Unterrichtsvorhaben EF-I</u></p> <p>Thema: Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten</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 <p>Zeitbedarf: ca. 9 Stunden</p>	<p><u>Unterrichtsvorhaben EF-II</u></p> <p>Thema: Grundlagen der objektorientierten Analyse, Modellierung und Implementierung anhand von einfachen Beispielkontexten</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: ca. 15 Stunden</p>
<p><u>Unterrichtsvorhaben EF-III</u></p> <p>Thema: Objektorientierte Analyse, Modellierung und Implementation anhand von komplexen Beispielkontexten</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 ➤ Syntax und Semantik einer Programmiersprache ➤ Analyse, Entwurf und Implementierung einfacher Algorithmen <p>Zeitbedarf: ca. 24 Stunden</p>	<p><u>Unterrichtsvorhaben EF-IV</u></p> <p>Thema: Such- und Sortieralgorithmen anhand kontextbezogener Beispiele</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> ➤ Argumentieren ➤ Modellieren ➤ Darstellen und Interpretieren ➤ Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> ➤ Daten und ihre Strukturierung ➤ Algorithmen <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> ➤ Algorithmen zum Suchen und Sortieren ➤ Analyse, Entwurf und Implementierung einfacher Algorithmen <p>Zeitbedarf: ca. 12 Stunden</p>

Einführungsphase

Unterrichtsvorhaben EF-V

Thema:

Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes

Zentrale Kompetenzen:

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatik, Mensch und Gesellschaft
- Informatiksysteme

Inhaltliche Schwerpunkte:

- Wirkungen der Automatisierung
- Geschichte der automatischen Datenverarbeitung
- Digitalisierung
- Datenschutz

Zeitbedarf: ca. 15 Stunden

Summe Einführungsphase: 75 Stunden

II) Qualifikationsphase (Q1 und Q2) - GRUNDKURS

Qualifikationsphase 1	
<p><u>Unterrichtsvorhaben Q1-I</u></p> <p>Thema: Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung</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 ➤ 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: ca. 9 Stunden</p>	<p><u>Unterrichtsvorhaben Q1-II</u></p> <p>Thema: Rekursive Programmierung an mathematischen und grafischen Beispielen</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 <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> ➤ Analyse, Entwurf und Implementierung von Algorithmen ➤ Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: ca. 9 Stunden</p>
<p><u>Unterrichtsvorhaben Q1-III</u></p> <p>Thema: Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen</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: ca. 15 Stunden</p>	<p><u>Unterrichtsvorhaben Q1-IV</u></p> <p>Thema: Suchen und Sortieren auf linearen Datenstrukturen</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 <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. 15 Stunden</p>

Qualifikationsphase 1

<p><u>Unterrichtsvorhaben Q1-V</u></p> <p>Thema: Modellierung und Nutzung 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. 21 Stunden</p>	<p><u>Unterrichtsvorhaben Q1-VI</u></p> <p>Thema: Sicherheit und Datenschutz 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>Summe Qualifikationsphase 1: 78 Stunden</p>	

Qualifikationsphase 2

<p><u>Unterrichtsvorhaben Q2-I</u></p> <p>Thema: Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen</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: ca. 24 Stunden</p>	<p><u>Unterrichtsvorhaben Q2-II</u></p> <p>Thema: Endliche Automaten und formale Sprachen</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> ➤ Argumentieren ➤ Modellieren ➤ Darstellen und Interpretieren ➤ Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> ➤ Endliche Automaten und formale Sprachen <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> ➤ Endliche Automaten ➤ Grammatiken regulärer Sprachen ➤ Möglichkeiten und Grenzen von Automaten und formalen Sprachen <p>Zeitbedarf: ca. 20 Stunden</p>
--	--

Qualifikationsphase 2

Unterrichtsvorhaben Q2-III

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: ca. 12 Stunden

Summe Qualifikationsphase 2: 56 Stunden

2.1.2 Konkretisierte Unterrichtsvorhaben

Im Folgenden sollen die im Unterkapitel 2.1.1 aufgeführten Unterrichtsvorhaben konkretisiert werden.

Hinweis:

Verbindliche Festlegung der Fachkonferenz:

Die Fachkonferenz der Heinrich-Böll-Gesamtschule 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 Heinrich-Böll-Gesamtschule. In diesen Bereichen sind Abweichungen von den vorgeschlagenen Vorgehensweisen möglich.

In der Einführungsphase und in der Qualifikationsphase wird die objektorientierte Programmiersprache Java verwendet. Dabei kommen die Entwicklungsumgebungen Greenfoot und BlueJ zum Einsatz.

- ◆ Greenfoot - <http://www.greenfoot.org/download> (Stand 01.09.2015)
- ◆ BlueJ - <http://www.bluej.org/> (Stand 01.09.2015)

In den Downloads der beiden Entwicklungsumgebungen ist jeweils das Java-Development-Kit (JDK) eingeschlossen. Dieses kann aber auch getrennt installiert werden. Es gibt entsprechende Versionen von Greenfoot und BlueJ ohne das JDK.

- ◆ <http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html#javasejdk> (Stand 01.09.2015)

In der Qualifikationsphase werden die Unterrichtsvorhaben unter Berücksichtigung der Vorgaben für das Zentralabitur Informatik in NRW konkretisiert. Diese sind zu beziehen unter der Adresse:

- ◆ <https://www.standardsicherung.schulministerium.nrw.de/abitur-gost/fach.php?fach=15> (Stand 01.09.2015)

In der Qualifikationsphase wird im Unterrichtsvorhaben Q1-V (Datenbanken) das Open-Source-Datenbanksystem MySQL verwendet. Als Frontend wird phpMyAdmin eingesetzt.

- ◆ <https://www.mysql.de/> (Stand 01.09.2015)
- ◆ <https://www.phpmyadmin.net/> (Stand 01.09.2015)

In der Qualifikationsphase werden im Unterrichtsvorhaben Q2-II (Endliche Automaten und formale Sprachen) die Werkzeuge Exorciser und JFlap verwendet.

- ◆ <http://www.swisseduc.ch/informatik/exorciser/> (Stand 01.09.2015)
- ◆ <http://www.jflap.org/> (Stand 01.09.2015)

I) Einführungsphase

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 sollten.

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. 9 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, 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>Beispiele:</p> <ul style="list-style-type: none"> ◆ <i>Textkodierung</i> Kodierung und Dekodierung von Texten (ASCII) ◆ <i>Bildkodierung</i> Kodierung von Bildinformationen in Raster- und Vektorgrafiken ◆ <i>Tonkodierung</i> Kodierung von Musik (MP3)
<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,</p>		<p>Beispiele:</p> <ul style="list-style-type: none"> ◆ <i>Rollenspiel zur Paketvermittlung im Internet</i> 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

<p>Zugriffsrechte, Client-Server)</p> <p>(c) Grundlagen der technischen Umsetzung von Rechnerkommunikation am Beispiel des Internets (z.B. Netzwerkadresse, Paketvermittlung, Protokoll)</p> <p>(d) Richtlinien zum verantwortungsvollen Umgang mit dem Internet</p>		<p>Routingtabelle. Mithilfe der Tabelle und einem Würfel wird entschieden, wie ein Paket weiter vermittelt wird.</p> <ul style="list-style-type: none"> ◆ <i>Simulationssoftware Filius</i> Schülerinnen und Schüler simulieren einfache Netze mit DHCP-, DNS- und Web-Server sowie Clients. Später werden dann Router hinzugenommen. <p>Medium:</p> <ul style="list-style-type: none"> ◆ <i>Video</i> Sendung mit der Maus http://www.wdrmaus.de/sachgeschichten/sachgeschichten/internet.php5 (Stand 02.09.2015)
<p>3. Aufbau informatischer Systeme</p> <p>(a) Identifikation typischer Komponenten informatischer Systeme und anschließende Beschränkung auf das Wesentliche, Herleitung der „Von-Neumann-Architektur“</p> <p>(b) Identifikation des EVA-Prinzips (Eingabe-Verarbeitung-Ausgabe) als Prinzip der Verarbeitung von Daten und Grundlage der „Von-Neumann-Architektur“</p>		<p>Material:</p> <ul style="list-style-type: none"> ◆ <i>Demonstrationshardware</i> Durch Demontage eines Demonstrationsrechners entdecken Schülerinnen und Schüler die verschiedenen Hardwarekomponenten eines Informatiksystems. Als Demonstrationsrechner bietet sich ein aus-rangierter Schulrechner an.

Unterrichtsvorhaben EF-II

Thema: Grundlagen der objektorientierten Analyse, Modellierung und Implementierung anhand von einfachen Beispielkontexten

Leitfrage: Wie lässt sich ein beispielhafter Gegenstandsbereich informatisch modellieren und im Form einer Simulation realisieren?

Vorhabenbezogene Konkretisierung:

Ein zentraler Bestandteil des Informatikunterrichts der EF ist die objektorientierte Programmierung. Dieses Unterrichtsvorhaben führt in die Grundlagen der Analyse, Modellierung und Implementation in diesem Kontext ein.

Dazu werden konkrete und anschauliche Grafiken analysiert und im Sinne des objektorientierten Paradigmas strukturiert. Dabei werden die grundlegenden Begriffe der Objektorientierung und Modellierungswerkzeuge wie Objektarten und Klassendiagramme eingeführt.

Im Anschluss wird mit der Realisierung erster Projekte mit Hilfe der Programmierumgebungen Greenfoot und BlueJ begonnen. Die von der Bibliothek vorgegebenen Klassen werden von Schülerinnen und Schülern in Teilen analysiert und entsprechende Objekte erprobt. Dazu muss der grundlegende Aufbau einer Java-Klasse erläutert und eingeführt werden, wobei zwischen Deklaration, Initialisierung und Methodenaufruf unterschieden werden muss.

Die selbst programmierten Projekte enthalten schon mehrere vorgegebene Klassen und im Wesentlichen nur eine selbst zu schreibende Klasse.

Zeitbedarf: ca. 15 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Identifikation von Objekten (a) An einem einfachen Beispiel werden Objekte im Sinne der objektorientierten Modellierung	Die Schülerinnen und Schüler ◆ ermitteln bei der Analyse einfacher Problem-	Beispiele: ◆ <i>Shapes (BlueJ)</i>

<p>eingeführt.</p> <p>(b) Manche Objekte sind prinzipiell typgleich und werden so zu einer Objektsorte bzw. Klasse zusammengefasst.</p> <p>(c) Klassen werden in Diagrammen visualisiert und mit sinnvollen Attributen und Methoden versehen.</p> <p>(d) Vertiefung: Modellierung weiterer Beispiele ähnlicher Art.</p>	<p>stellungen Objekte, ihre Eigenschaften und ihre Operationen (M)</p> <ul style="list-style-type: none"> ◆ modellieren Klassen mit ihren Attributen und Methoden (M), ◆ stellen Attribute und Methoden in einem Klassendiagramm dar (D), ◆ ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen zu (M), 	<p>Mit Dreiecken, Quadraten und Kreisen werden Figuren gezeichnet.</p> <p>(David J. Barnes, Michael Kölling; Java Lernen mit BlueJ; PEARSON Studium)</p> <ul style="list-style-type: none"> ◆ <i>Auto (BlueJ)</i> Ein Auto mit Attributen Tankinhalt, Kilometerstand und Verbrauch sowie Methoden mit Parametern
<p>2. Implementation einfacher Klassen mit BlueJ</p> <p>(a) Grundaufbau einer Java</p> <p>(b) Deklaration und Initialisierung von Objekten</p> <p>(c) Konstruktoren</p> <p>(d) Einfache Methodenaufrufe</p> <p>(e) Primitive Datentypen als Attribute, Parameter und Variablen</p> <p>(f) Bedingte Anweisungen (if-Abfragen)</p>	<ul style="list-style-type: none"> ◆ implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), ◆ interpretieren Fehlermeldungen und korrigieren den Quellcode (I), ◆ stellen den Zustand eines Objektes dar (D), ◆ analysieren und erläutern einfache Algorithmen und Programme (A), ◆ entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch (PAP) dar (M), 	
<p>3. Einfache Greenfoot-Szenarien</p> <p>(a) Analyse von vorgegebenen Klassen</p> <p>(b) Grundlagen der Vererbung</p> <p>(c) Abfrage von Zuständen</p>	<ul style="list-style-type: none"> ◆ implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I), ◆ implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I). 	<p>Beispiele:</p> <ul style="list-style-type: none"> ◆ Whombats-Szenario Hindernisse, Gegenstände aufsammeln, Erkennen von Hindernissen und Gegenständen ◆ <i>Weitere Greenfoot-Senarios</i> (Website Greenfoot.org)

Unterrichtsvorhaben EF-III

Thema: Objektorientierte Analyse, Modellierung und Implementation anhand von komplexen Beispielkontexten

Leitfragen: Wie lassen sich komplexere Datenflüsse und Beziehungen zwischen Objekten und Klassen realisieren? Wie kann man Tastatur und Mauseingaben realisieren?

Vorhabenbezogene Konkretisierung:

Aufbauend auf dem Unterrichtsvorhaben II wird die Objektorientierung vertieft und erweitert.

Es werden nun Projekte in Greenfoot und BlueJ realisiert, bei denen mehr als eine Klasse implementiert werden muss. Die Beziehungen zwischen den Klassen werden analysiert und in Klassendiagrammen dargestellt. Insbesondere wird die Vererbung vertieft.

Attribute, Parameter und Variablen können nun auch Objektreferenzen sein. In diesem Zusammenhang muss zwischen Objektreferenzen und Objekten genau unterschieden werden. Nachdem im Unterrichtsvorhaben EF-II die if-Abfrage als eine erste Kontrollstruktur eingeführt wurde, kommen nun die Schleifenstrukturen hinzu.

Bei den Greenfoot-Projekten sollen zunächst Spiele und Simulationen realisiert werden, bei denen der Benutzer per Maus- oder Tastatureingabe steuernd eingreifen kann. Dementsprechend werden Möglichkeiten der Interaktion bei Greenfoot-Szenarien in diesem Unterrichtsvorhaben behandelt. Bei BlueJ-Projekten liegt der Schwerpunkt auf der Behandlung linearer Datensammlungen.

Zeitbedarf: ca. 24 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Entwicklung von Spielen oder Simulationen mit Interaktion zwischen unterschiedlichen Objekten in Greenfoot	Die Schülerinnen und Schüler <ul style="list-style-type: none">◆ analysieren und erläutern eine	Beispiele: <ul style="list-style-type: none">◆ <i>Schatzräuber (Greenfoot)</i>

<ul style="list-style-type: none"> (a) Schrittweise Implementation der Projekte (b) Kontinuierliches Bewegen eines Objekts mit der while-Schleife und der do-while-Schleife (c) Definiertes Bewegen eines Objekts mit der for-Schleife (d) Dokumentation der Klassen des Projekts (e) Kollisionsabfragen (f) Verwendung von Objektreferenzen 	<p>objektorientierte Modellierung (A),</p> <ul style="list-style-type: none"> ◆ stellen die Kommunikation zwischen Objekten in Diagrammen 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), ◆ modellieren Klassen unter Verwendung von Vererbung (M), ◆ implementieren Klassen auch unter Nutzung dokumentierter Klassenbibliotheken (I), 	<p>In einem Szenario mit Wänden und Schätzen versucht ein Abenteurer Schätze aufzusammeln. Ein Wächter versucht ihn daran zu hindern, indem er ihn einfängt. Beide Figuren werden mit der Tastatur gesteuert.</p> <ul style="list-style-type: none"> ◆ <i>Erweiterungen des Schatzräuber-Szenarios</i> (z.B. Maurer und Abreißer, oder auch Tunnel und Helfer) ◆ <i>Rotation (Greenfoot)</i> Verwandt mit dem Schiebepuzzle. Zahlen müssen durch Rotation von Blöcken aus jeweils 4 Feldern sortiert werden. Das Spiel wird mit der Maus gesteuert. <p>Weitere Beispiel alternativ:</p> <ul style="list-style-type: none"> ◆ <i>Little Crab-Szenario (Greenfoot)</i> (Michael Kölling, Einführung in Java mit Greenfoot, PEARSON) ◆ <i>Räuber-Beute (Greenfoot)</i> Erstellung einer Räuber-Beute Simulation
<p>2. Vertiefung der Vererbung</p> <ul style="list-style-type: none"> (a) Analyse und Erläuterung einer Basis-Version einer Klasse (b) Realisierung von Erweiterungen zur Basisklasse mit Vererbung (c) Verallgemeinerung und Reflexion des Prinzips der Vererbung am Beispiel der Spezialisierung (d) Überschreiben von Methoden (e) abstrakte Oberklassen 	<ul style="list-style-type: none"> ◆ testen Programme schrittweise anhand von Beispielen (I), ◆ interpretieren Fehlermeldungen und korrigieren den Quellcode (I), ◆ modifizieren einfache Algorithmen und Programme (I), ◆ stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), 	<p>Beispiele:</p> <ul style="list-style-type: none"> ◆ <i>Krümelmonster (BlueJ)</i> Erstellung eines einfachen Würfelspiels mit Keksen als Einsatz ◆ <i>Wortsalat (BlueJ)</i> Die Buchstaben eines Wortes sind durcheinander geraten und das Wort muss erraten werden. ◆ <i>Hangman (BlueJ)</i>
<p>3. Entwicklung von BlueJ-Projekten mit linearen Datensammlungen fester Größe.</p> <ul style="list-style-type: none"> (a) Deklaration und Initialisierung von eindimensionalen Feldern (b) Zugriff auf die Elemente eines eindimensionalen Feldes (c) Anzahl der Elemente eines eindimensionalen Feldes (d) Darstellung eindimensionaler Felder in 		

<p>Implementationsdiagrammen</p> <p>(e) Eindimensionale Felder als Attribute, Variablen, Parameter und Rückgaben von Methoden</p> <p>(f) Die Klasse Math</p> <p>(g) Die Klasse String</p>	<p>◆ dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D).</p>	<p>Implementierung des bekannten Spiels</p> <p>◆ <i>Biester (Greenfoot)</i></p> <p>Die Biester suchen und fressen herabrieselndes Plankton am Grunde des Meeres. Die Biester können ihr Suchverhalten verändern (genetischer Algorithmus).</p>
---	--	--

Unterrichtsvorhaben EF-IV

Thema: Such- und Sortieralgorithmen anhand kontextbezogener Beispiele

Leitfrage: 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 wird.

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 anschließende 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: ca. 12 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>Beispiele:</p> <ul style="list-style-type: none"> ◆ <i>Sortieren mit Waage</i> 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>Materialien:</p> <ul style="list-style-type: none"> ◆ <i>Computer Science Unplugged – Sorting Algorithms</i> http://csunplugged.org/sorting-algorithms/ (Stand 04.09.2015) ◆ <i>Registerkartenmodell</i> ◆ <i>Schülergruppe</i> ◆ <i>Algorithmus der Woche</i> http://www-i1.informatik.rwth-aachen.de/~algorithmus/index.php ◆ <i>Mathe-Prisma</i> http://www.matheprisma.uni-wuppertal.de/

<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 Rechenzeit und des Speicherplatzbedarfs</p> <p>(f) Analyse des weiteren Sortieralgorithmus (sofern nicht in Sequenz 1 und 2 bereits geschehen)</p>		<p>Beispiele:</p> <ul style="list-style-type: none"> ◆ Sortieren durch Auswählen ◆ Sortieren durch Vertauschen ◆ <i>Quicksort</i> Ist als Beispiel für einen Algorithmus nach dem Prinzip „Teile und Herrsche“ gut zu behandeln. Kenntnisse in rekursiver Programmierung sind nicht erforderlich, da eine Implementierung nicht angestrebt wird. <p>Materialien:</p> <ul style="list-style-type: none"> ◆ <i>Computer Science Unplugged – Sorting Algorithms</i> http://csunplugged.org/sorting-algorithms/ (Stand 04.09.2015) ◆ <i>Mathe-Prisma</i> http://www.matheprisma.uni-wuppertal.de/
<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) Effizienzbetrachtung zur binären Suche</p>		<p>Beispiel:</p> <ul style="list-style-type: none"> ◆ <i>Simulationsspiel zur binären Suche nach Tischtennisbällen</i> Mehrere Tischtennisbälle sind nummeriert, sortiert und unter Bechern verdeckt. Mithilfe der binären Suche kann sehr schnell ein bestimmter Tischtennisball gefunden werden. <p>Materialien:</p> <ul style="list-style-type: none"> ◆ <i>Computer Science Unplugged – Sorting</i>

		<i>Algorithms</i> http://csunplugged.org/sorting-algorithms/ (Stand 04.09.2015) ◆ <i>Mathe-Prisma</i> http://www.matheprisma.uni-wuppertal.de/
--	--	--

Unterrichtsvorhaben EF-V

Thema: Die digitale Welt und die Grundlagen des Datenschutzes

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

Vorhabenbezogene Konkretisierung:

Dieses Unterrichtsvorhaben stellt den Abschluss der Einführungsphase dar. Die erste Unterrichtssequenz („Informatiker verändern die Welt – Auswirkungen der Informationstechnologie auf Mensch und Gesellschaft“) kann von den Schülerinnen und Schülern in Form von Referaten oder in Kleingruppenarbeit präsentiert werden.

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

Zeitbedarf: ca. 15 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Informatiker verändern die Welt – Auswirkungen der Informationstechnologie auf Mensch und Gesellschaft</p> <p>(a) Von analog zu digital (b) Die Binäre Welt (c) Der von-Neumann-Rechner (d) Von der Schrift zum Smartphone (e) Leben in der digitalen Welt (f) Automatisierung (g) Was macht Informatik?</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ◆ stellen ganze Zahlen und Zeichen in Binärcodes dar (D), ◆ interpretieren Binärcodes als Zahlen und Zeichen (D), ◆ bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A), ◆ erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A), ◆ nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K) 	<p>Medien:</p> <ul style="list-style-type: none"> ◆ <i>Schulbuch</i> (Informatik 1 und 2, Schöningh) ◆ <i>Weitere Bücher</i> (Schulbibliothek) ◆ <i>Internet</i> <p>Materialien:</p> <ul style="list-style-type: none"> ◆ (a) – Informatik 1, S. 34-35 ◆ (b) – Informatik 1, S. 70-77 ◆ (c) – Informatik 1, S. 128-129 ◆ (d) – Informatik 1, S. 162-169 ◆ (e) – Informatik 1, S. 188-194 ◆ (f) – Informatik 3, S. 56-57, S. 75-81 ◆ (g) – Informatik 1, S. 6-17 ◆ <i>Informatik und ethische Aspekte</i> Informatik 1, S 6-17, u.a. Informatik 3, S 60-65
<p>2. Datenschutz – Der gläserne Mensch</p> <p>(a) Erarbeitung grundlegender Begriffe des Datenschutzes (b) Problematisierung und Anknüpfung an die Lebenswelt der Schülerinnen und Schüler (c) Diskussion und Bewertung von Fallbeispielen aus dem Themenbereich „Datenschutz“</p>		<p>Medien:</p> <ul style="list-style-type: none"> ◆ <i>Schulbuch</i> (Informatik 1 und 2 Schöningh) ◆ <i>Weitere Bücher</i> (Schulbibliothek) ◆ <i>Internet</i> <p>Materialien:</p>

		<ul style="list-style-type: none">◆ <i>Der gläserne Deutsche</i> Informatik 3 S. 54-55◆ <i>Der gläserne Mensch</i> Informatik 3, S. 66 -67◆ <i>Datenschutz als Grundrecht?</i> Informatik 3, S. 68-69◆ <i>Datenschutz konkret – NSA und Social Media</i> Informatik 3, S. 74◆ <i>Datenschutzgesetz</i> http://www.gesetze-im-internet.de/bundesrecht/bdsg_1990/gesamt.pdf
--	--	--

II) Qualifikationsphase

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).

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?

Vorhabenbezogene 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. Der Nachrichtenaustausch zwischen verschiedenen Objekten wird verdeutlicht, indem die Kommunikation zwischen zwei ausgewählten Objekten grafisch dargestellt wird. In diesem Zusammenhang wird das Nachrichtenkonzept der objektorientierten Programmierung wiederholt.

Zeitbedarf: ca. 9 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</p> <p>(b) Analyse der Modellierung (Implementationsdiagramm)</p> <p>(c) Erweiterung der Modellierung im Implementationsdiagramm (Vererbung, abstrakte Klasse)</p> <p>(d) Kommunikation zwischen mindestens zwei Objekten (grafische Darstellung)</p> <p>(e) Dokumentation von Klassen</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ◆ analysieren und erläutern objektorientierte Modellierungen (A), ◆ beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), ◆ modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), ◆ ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), ◆ modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), 	<p>Beispiele (alternativ):</p> <ul style="list-style-type: none"> ◆ <i>Wetthüpfen</i> Zusatzmaterial zum Kernlehrplan, Unterrichtsvorhaben Q1-I.1/Wetthuepfen http://www.schulentwicklung.nrw.de/materialdatenbank/nutzersicht/materialeintrag.php?matId=4353 (Stand 05.09.2015) 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. ◆ <i>Banksimulation</i> Eine Simulation mit

<p>(f) Implementierung der Anwendung oder von Teilen der Anwendung</p>	<ul style="list-style-type: none"> ◆ implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), ◆ nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), ◆ wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I), ◆ interpretieren Fehlermeldungen und korrigieren den Quellcode (I), ◆ stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), ◆ dokumentieren Klassen (D), ◆ stellen die Kommunikation zwischen Objekten grafisch dar (D). 	<ul style="list-style-type: none"> - Geldautomaten, - Konten, - Chipkarten, - Kunden, - Schaltern - ... <p>Je nach Umfang teilweise mit vorgegebenen Klassen</p> <ul style="list-style-type: none"> ◆ <i>Bruchrechner mit GUI</i> Grundrechenarten mit Brüchen. Eine GUI zur Darstellung der Brüche wird vorgegeben.
--	---	---

Unterrichtsvorhaben Q1-II

Thema: Rekursive Programmierung an mathematischen und grafischen Beispielen

Leitfragen: Wie können mathematische Folgen rekursiv programmiert werden. Wie können Grafiken in der Art von Lindenmayer-Systemen rekursiv programmiert und dargestellt werden?

Vorhabenbezogene Konkretisierung:

Die Schülerinnen und Schülern erläutern auf einer ersten Ebene anhand einfacher mathematischer Folgen die Grundidee der Rekursion

und vergleichen und bewerten rekursive und iterative Ansätze zur Berechnung der gleichen Folgen. Sie implementieren einfache rekursive Methoden zur Berechnung einer Folge. Die Schülerinnen und Schüler machen vergleichende Aussagen über die Laufzeit und den Speicherplatzbedarf iterativer und rekursiver Methoden.

Auf einer zweiten Ebene erstellen die Schülerinnen und Schüler rekursive Programme zur Darstellung von einfachen Fraktalen (Lindenmayer-Systemen).

In diesem Unterrichtsvorhaben ergeben sich vielfältige Verknüpfungen zu dem Fach Mathematik.

Zeitbedarf: ca. 9 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Rekursive und iterative Berechnung von mathematischen Folgen</p> <p>(a) Analyse und Berechnung mathematischer Folgen</p> <p>(b) Implementation rekursiver Methoden zur Berechnung von Folgen</p> <p>(c) Analyse von rekursiven Methoden unter Laufzeit- und Speicherplatzaspekten</p> <p>(d) Analyse einfacher Lindenmayer-Systeme</p> <p>(e) Implementation einfacher Fraktale mithilfe einer Turtle-Grafik</p> <p>(f) Abbruchbedingungen für die Methoden und Aussagen über das Laufzeitverhalten</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ◆ analysieren und erläutern Algorithmen und Programme (A), ◆ stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D), ◆ entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M), ◆ implementieren iterative und rekursive Algorithmen (I), ◆ beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A). 	<p>Beispiele für mathematische Folgen:</p> <ul style="list-style-type: none"> ◆ <i>Fakultät</i> Die Schülerinnen und Schüler lernen die iterative ($n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$) und die rekursive Form ($n! = n \cdot (n-1)!$, $0! = 1$) kennen und implementieren beide Formen. ◆ <i>Die Fibonacci-Folge</i> $f_n = f_{n-2} + f_{n-1}, \quad f_1 = 1, \quad f_2 = 1$ ◆ Binomialkoeffizienten $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}, \quad \binom{n}{0} = 1, \quad \binom{n}{n} = 1$ <p>Beispiele für Fraktale (Lindenmayer-Systeme)</p> <ul style="list-style-type: none"> ◆ Kochkurve und Kochflocke

		<ul style="list-style-type: none">◆ Schafgarbe◆ Barnsleys Farn◆ Pythagorasbaum
--	--	--

Unterrichtsvorhaben Q1-III

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

Leitfragen: Wie können beliebig viele linear angeordnete Daten im Anwendungskontext verwaltet werden?

Vorhabenbezogene Konkretisierung:

Nach Analyse einer Problemstellung in einem geeigneten Anwendungskontext, in dem Daten nach dem First-In-First-Out-Prinzip verwaltet werden, werden der Aufbau von Schlangen am Beispiel dargestellt und die Operationen der Klasse Queue erläutert. Anschließend werden für die Anwendung notwendige Klassen modelliert und implementiert. Eine Klasse für eine den Anforderungen der Anwendung entsprechend 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 und Listen vertieft. Modellierungen werden dabei in Entwurfs- und Implementationsdiagrammen dargestellt.

Zeitbedarf: ca. 15 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. 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 Queue</p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Queue</p> <p>(d) Erstellen eines Implementationsdiagramms für die Anwendung</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ◆ erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), ◆ analysieren und erläutern Algorithmen und Programme (A), ◆ beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), ◆ ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nicht-lineare Datensammlungen zu (M), ◆ ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), ◆ modifizieren Algorithmen und Programme (I), ◆ implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), ◆ nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), ◆ interpretieren Fehlermeldungen und korrigieren den Quellcode (I), ◆ testen Programme systematisch anhand von Beispielen (I), 	<p>Beispiele:</p> <ul style="list-style-type: none"> ◆ <i>Druckerwarteschlange</i> Die Druckjobs auf einem lokalen Computer sollen verwaltet werden. Dazu gibt es eine Warteschlange, welche die Druckjobs nach dem First-In-First-Out-Prinzip speichert. ◆ <i>Patientenwarteschlange</i> Zusatzmaterial zum Kernlehrplan, Unterrichtsvorhaben Q1-II.1\Wartezimmer http://www.schulentwicklung.nrw.de/material/datenbank/nutzersicht/materialeintrag.php?matId=4353 (Stand 05.09.2015) 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. Die Simulationsanwendung stellt eine GUI zur Verfügung, legt ein Wartezimmer an und steuert die Abläufe. Wesentlicher Aspekt des Projektes ist die Modellierung des Wartezimmers mithilfe der Klasse Queue. Anschließend wird der Funktionsumfang der Anwendung erweitert: Patienten können sich zusätzlich in die Warteschlange zum Blutdruckmessen einreihen. Objekte werden von zwei Schlangen verwaltet. <p>Materialien:</p>

	<ul style="list-style-type: none"> ◆ stellen lineare und nichtlinear Strukturen grafisch dar und erläutern ihren Aufbau (D). 	<ul style="list-style-type: none"> ◆ GUI-DaLi Eine GUI zur Darstellung von Listen.
<p>2. Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse Stack</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse Stack</p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Stack</p>		<p>Beispiele:</p> <ul style="list-style-type: none"> ◆ <i>Heftstapel</i> In einem Heftstapel soll das Heft einer Schülerin gefunden werden. ◆ <i>Kisten stapeln</i> In einem Stapel nummerierter Kisten soll eine bestimmte Kiste gefunden und an einen Kunden geliefert werden. Dazu müssen Kisten auf verschiedene Stapel gestapelt und wieder zurückgestellt werden. ◆ <i>Spielkartenstack</i> Implementation eines Kartenschuhs, wie er in Spielkasinos verwendet wird und Darstellung mithilfe der GUI-DaLi ◆ <i>Türme von Hanoi</i> Implementation des Spiels mit 3 Stacks und Darstellung mithilfe der GUI-DaLi <p>Materialien:</p> <ul style="list-style-type: none"> ◆ <i>GUI-DaLi</i> Eine GUI zur Darstellung von Listen
<p>3. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse List</p> <p>(a) Erarbeitung der Vorteile der Klasse List im Gegensatz zu den bereits bekannten linearen Strukturen</p>		<p>Beispiel:</p> <ul style="list-style-type: none"> ◆ <i>Abfahrtslauf</i> Zusatzmaterial zum Kernlehrplan, Unterrichtsvorhaben Q1-II.2\Abfahrtslauf http://www.schulentwicklung.nrw.de/materialdatenbank/nutzersicht/materialeintrag.php?

<p>(b) Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse List</p>		<p>matId=4353 (Stand 05.09.2015) 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>4. Vertiefung – Anwendungen von Listen, Stapeln oder Schlangen in mindestens einem weiteren Kontext</p>		<p>Beispiele:</p> <ul style="list-style-type: none"> ◆ <i>Priority-Queue</i> Ausgehend von dem Beispiel Wartezimmer wird eine Klasse PriorityQueue, welche Prioritätenobjekte verwaltet, modelliert und implementiert. Intern verwendet die Priority-Queue eine Liste zur Verwaltung der Objekte. ◆ <i>Ordered-List</i> Ausgehend von dem Beispiel Abfahrtslauf wird eine Klasse Orderd-List modelliert und implementiert. Intern verwendet die Ordered-List eine Liste zur Verwaltung der Objekte. Alternativ: Eine Kontaktverwaltung ◆ <i>Ring</i> Ein Kartenspiel soll gemischt werden. Dies ist besonders einfach in einer Ringstruktur zu realisieren. Intern verwendet der Ring eine Liste. ◆ <i>Skispringen</i> Zusatzmaterial zum Kernlehrplan, Unterrichtsvorhaben Q1-II.3\Skispringen http://www.schulentwicklung.nrw.de/material/datenbank/nutzersicht/materialeintrag.php?

[matId=4353](#) (Stand 05.09.2015)

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 eine Punktzahl und wird nach der Gesamtpunktzahl aus beiden Durchgängen in die endgültige Rangliste eingeordnet.

◆ *Terme in Postfix-Notation*

Zusatzmaterial zum Kernlehrplan,
Unterrichtsvorhaben Q1-II.3\Postfix

<http://www.schulentwicklung.nrw.de/materialdatenbank/nutzersicht/materialeintrag.php?matId=4353> (Stand 05.09.2015)

Die sog. UPN (Umgekehrt-Polnische-Notation) bzw. Postfix-Notation 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.

◆ *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.) 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. Eine Darstellung ist mithilfe der GUI-DaLI möglich.

- ◆ *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 „Heranfahen“ 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.

Unterrichtsvorhaben Q1-IV

Thema: Suchen und Sortieren auf linearen Datenstrukturen

Leitfragen: Wie kann man gespeicherte Informationen günstig (wieder-)finden?

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 Sortierverfahren entwickelt und implementiert (ebenfalls für lineare Listen und Felder). Hierbei soll auch ein rekursives Sortierverfahren entwickelt werden. Die Implementierungen 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: ca. 15 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Suchen von Daten in Listen und Arrays (a) Lineare Suche in Listen und in Arrays (b) Binäre Suche in Arrays als Beispiel für rekursives Problemlösen (c) Untersuchung der beiden Suchverfahren	Die Schülerinnen und Schüler <ul style="list-style-type: none">◆ analysieren und erläutern Algorithmen und Programme (A),◆ beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),	Beispiele: <ul style="list-style-type: none">◆ <i>Karteiverwaltung</i> Zusatzmaterial zum Kernlehrplan, Unterrichtsvorh. Q1-III.1\Karteiverwaltung http://www.schulentwicklung.nrw.de/material/datenbank/nutzersicht/materialeintrag.php?

<p>hinsichtlich ihrer Effizienz (Laufzeitverhalten, Speicherbedarf)</p>	<ul style="list-style-type: none"> ◆ beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A), ◆ entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M), ◆ modifizieren Algorithmen und Programme (I), ◆ implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), ◆ implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I), ◆ nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), ◆ interpretieren Fehlermeldungen und korrigieren den Quellcode (I), ◆ testen Programme systematisch anhand von Beispielen (I), ◆ stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D). 	<p>matId=4353 (Stand 05.09.2015)</p> <p>Für ein Adressverwaltungsprogramm soll eine Methode zum Suchen einer Adresse geschrieben werden.</p> <ul style="list-style-type: none"> ◆ <i>Bundesjugendspiele</i> Zusatzmaterial zum Kernlehrplan, Unterrichtsvorh. Q1-III.1\Bundesjugendspiele http://www.schulentwicklung.nrw.de/materialdatenbank/nutzersicht/materialeintrag.php?matId=4353 (Stand 05.09.2015) <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>2. Sortieren in Listen und Arrays – Entwicklung und Implementierung von iterativen und rekursiven Sortierverfahren</p> <p>(a) Entwicklung und Implementierung eines einfachen Sortierverfahrens für eine Liste</p> <p>(b) Implementierung eines einfachen Sortierverfahrens für ein Feld</p> <p>(c) Entwicklung eines rekursiven Sortierverfahrens für ein Feld (z.B. Sortieren durch Mischen)</p>		<p>Beispiele:</p> <ul style="list-style-type: none"> ◆ <i>Karteiverwaltung (s.o)</i> ◆ <i>Bundesjugendspiele (s.o)</i>

<p>3. Untersuchung der Effizienz der Sortierverfahren „Sortieren durch direktes Einfügen“ und „Quicksort“ auf linearen Listen</p> <p>(a) Grafische Veranschaulichung der Sortierverfahren</p> <p>(b) Untersuchung der Anzahl der Vergleichsoperationen und des Speicherbedarf bei beiden Sortierverfahren</p> <p>(c) Beurteilung der Effizienz der beiden Sortierverfahren</p>		<p>Beispiele:</p> <ul style="list-style-type: none"> ◆ <i>Karteiverwaltung (s.o)</i> ◆ <i>Bundesjugendspiele (s.o)</i>
---	--	---

Unterrichtsvorhaben Q1-V

Thema: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

Leitfragen: Wie können Fragestellungen mithilfe 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.

Zeitbedarf: ca. 21 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 (SELECT (DISTINCT) ...FROM, WHERE, AND, OR, NOT) auf einer Tabelle 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ◆ erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A), ◆ analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A), ◆ analysieren und erläutern eine Datenbankmodellierung (A), ◆ erläutern die Eigenschaften normalisierter Datenbankschemata (A), ◆ bestimmen Primär- und Sekundärschlüssel (M), ◆ ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, 	<p>Beispiel:</p> <ul style="list-style-type: none"> ◆ <i>VideoCenter</i> http://videocenter.schule.de/ <p>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 (Stand 05.09.2015) 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</p>

<ul style="list-style-type: none"> ▪ Analyse und Erarbeitung von SQL-Abfragen auf einer und mehrerer Tabelle zur Beantwortung der Fragestellungen (JOIN, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, <>, >, <, >=, <=, LIKE, BETWEEN, IN, IS NULL) <p>(c) Vertiefung an einem weiteren Datenbankbeispiel</p>	<p>Relationen und Kardinalitäten (M),</p> <ul style="list-style-type: none"> ◆ 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), 	<p>Folgenden beschriebenen Durchführung verwendet werden kann.</p> <p>Bitte eine Email an mailto:penon@bics.be.schule.de, um die Zugangskennungen für die Datenbank zu erhalten.</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 	<ul style="list-style-type: none"> ◆ verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I), ◆ ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D), ◆ stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D), ◆ überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D). 	<p>Beispiele:</p> <ul style="list-style-type: none"> ◆ <i>CD-Datenbank</i> Anhand der CD-Datenbank werden die erste bis dritte Normalform und weitere erläutert. https://de.wikipedia.org/wiki/Normalisierung_%28Datenbank%29 ◆ <i>Fahrradverleih</i> Der Fahrradverleih BTR (BikesToRent) verleiht unterschiedliche Typen von Fahrrädern diverser Firmen an seine Kunden. Die Kunden sind bei BTR registriert (Name, Adresse, Telefon). BTR kennt von den Fahrradfirmen den Namen und die Telefonnummer. Kunden von BTR können CityBikes, Treckingräder und Mountainbikes ausleihen. ◆ <i>Reederei</i> Die Datenverwaltung einer Reederei soll in einem Datenbanksystem umgesetzt werden. Ausgehend von der Modellierung soll mithilfe eines ER-Modells und eines Datenbankschemas dieser erste Entwurf normalisiert und in einem Datenbanksystem umgesetzt

<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) 		<p>werden. Es schließen sich diverse SQL-Abfragen an, wobei auf die Relationenalgebra eingegangen wird.</p> <ul style="list-style-type: none"> ◆ <i>Buchungssystem</i> In dem Online-Buchungssystem einer Schule können die Lehrerinnen und Lehrer Medienräume, Beamer, Laptops, Kameras, usw. für einen bestimmten Zeitpunkt buchen, der durch das 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/ (Stand 05.09.2015) findet man ein freies Online-Buchungssystem inklusive einer Demo, anhand derer man erläutern kann, worum es in dem Projekt geht. ◆ <i>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 2014/2015 in der Jahrgangsstufe 9 im Differenzierungsbereich im Fach Physik 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.
--	--	--

Unterrichtsvorhaben Q1-VI

Thema: Sicherheit und Datenschutz in Netzstrukturen

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

Vorhabenbezogene Konkretisierung:

Anschließend an das vorhergehende Unterrichtsvorhaben zum Thema Datenbanken werden der Datenbankzugriff aus dem Netz, Topologien von Netzwerken, eine Client-Server-Struktur, das TCP/IP-Schichtmodell 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.

Zeitbedarf: ca. 9 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Daten in Netzwerken und Sicherheitsaspekte in Netzen sowie beim Zugriff auf Datenbanken (a) Beschreibung eines Datenbankzugriffs im Netz anhand eines Anwendungskontextes und einer Client-Server-Struktur zur Klärung der Funktionsweise eines Datenbankzugriffs (b) Netztopologien als Grundlage von Client-Server-Strukturen und TCP/IP-Schichtenmodell als Beispiel für eine Paketübermittlung in einem Netz	Die Schülerinnen und Schüler <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	Materialien: <ul style="list-style-type: none">◆ Zusatzmaterial zum Kernlehrplan, UnterrichtsvorhabenQ1-V.1 - Zugriff auf Daten in Netzwerken http://www.schulentwicklung.nrw.de/materialdatenbank/nutzersicht/materialeintrag.php?matId=4353 (Stand 05.09.2015)

<p>(c) Vertraulichkeit, Integrität, Authentizität in Netzwerken sowie symmetrische und asymmetrische kryptografische Verfahren (Cäsar-, Vigenère-, RSA-Verfahren) als Methoden Daten im Netz verschlüsselt zu übertragen</p>	<p>von Informatiksystemen, die Sicherheit von Informatiksystemen sowie die Einhaltung der Datenschutzbestimmungen und des Urheberrechts (A),</p>	
<p>2. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht</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>Materialien:</p> <ul style="list-style-type: none"> ◆ Zusatzmaterial zum Kernlehrplan, Unterrichtsvorhaben Q1-V.2 - Datenschutz Beim VideoCenter, Materialblatt - Datenschutzgesetz http://www.schulentwicklung.nrw.de/materialdatenbank/nutzersicht/materialeintrag.php?matId=4353 (Stand 05.09.2015)

Unterrichtsvorhaben Q2-I

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

Leitfragen: Wie können Daten im Anwendungskontext mithilfe 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 Bauminhalte 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.

Zeitbedarf: ca. 24 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“ und „Teilen und Herrschen“ (M), ◆ implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), 	<p>Beispiele:</p> <ul style="list-style-type: none"> ◆ <i>Termbaum</i> Der Aufbau von Termen wird mit Hilfe von binären Baumstrukturen verdeutlicht. ◆ <i>Ahnenbaum</i> Die binäre Baumstruktur ergibt sich daraus, dass jede Person genau einen Vater und eine Mutter hat. <p>Weitere Beispiele für Anwendungskontexte für binäre Bäume:</p> <ul style="list-style-type: none"> ◆ <i>Suchbäume</i> (zur sortierten Speicherung von Daten) 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.) ◆ <i>Entscheidungsbäume</i> 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. ◆ <i>Codierungsbäume</i> für Codierungen, deren Alphabet aus genau

	<ul style="list-style-type: none"> ◆ 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>zwei Zeichen besteht Zusatzmaterial zum Kernlehrplan, Unterrichtsvorhaben Q2-I.1\Morsebaum http://www.schulentwicklung.nrw.de/materialdatenbank/nutzersicht/materialeintrag.php?matId=4354 (Stand 06.09.2015)</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>2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse BinaryTree</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms</p> <p>(c) Erarbeitung der Klasse BinaryTree und beispielhafte Anwendung der Operationen</p> <p>(d) Implementierung der Anwendung oder von Teilen der Anwendung</p> <p>(e) Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf</p>		<p>Beispiel:</p> <ul style="list-style-type: none"> ◆ <i>Informatikerbaum als binärer Baum</i> Zusatzmaterial zum Kernlehrplan, Unterrichtsvorhaben Q2-I.2\ BinaryTree Informatiker http://www.schulentwicklung.nrw.de/materialdatenbank/nutzersicht/materialeintrag.php?matId=4354 (Stand 06.09.2015) <p>In einem binären Baum 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>3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse BinarySearchTree</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms, grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften</p> <p>(c) Erarbeitung der Klasse BinarySearchTree und Einführung des Interface Comparable-Content zur Realisierung einer geeigneten Ordnungsrelation</p> <p>(d) Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums</p>		<p>Beispiel:</p> <ul style="list-style-type: none"> ◆ <i>Informatikerbaum als Suchbaum</i> Zusatzmaterial zum Kernlehrplan, Unterrichtsvorhaben Q2-I.3\ Informatiker_BinarySearchTree http://www.schulentwicklung.nrw.de/material/datenbank/nutzersicht/materialeintrag.php?matId=4354 (Stand 06.09.2015) <p>In einem binären Suchbaum 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

4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen

Beispiel:

- ◆ *Codierungsbäume (s.o.) oder Huffman-Codierung*
- ◆ Buchindex
Zusatzmaterial zum Kernlehrplan,
Unterrichtsvorhaben Q2-I.4\
BinarySearchtree_BuchIndex
<http://www.schulentwicklung.nrw.de/materialdatenbank/nutzersicht/materialeintrag?matId=4354> (Stand 06.09.2015)
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 Buchindex als Suchbaum (Objekt der Klasse BinarySearchTree) 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.)
- ◆ *Entscheidungsbäume (s.o.)*
- ◆ *Termbaum (s.o.)*
- ◆ *Ahnenbaum (s.o.)*

Unterrichtsvorhaben Q2-II

Thema: Endliche Automaten und formale Sprachen

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.

Zeitbedarf: ca. 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, 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), 	<p>Beispiele:</p> <ul style="list-style-type: none"> ◆ Cola-Automat, ◆ Geldspielautomat, ◆ Roboter, ◆ Zustandsänderung eines Objekts „Auto“, ◆ Akzeptor für bestimmte Zahlen, ◆ Akzeptor für Teilwörter in längeren Zeichenketten, ◆ Akzeptor für Terme <p>Materialien:</p> <p>Zusatzmaterial zum Kernlehrplan, Unterrichtsvorhaben Q2-II.1 http://www.schulentwicklung.nrw.de/materialdatenbank/nutzersicht/materialeintrag.php?matId=4354 (Stand 06.09.2015)</p>
<p>2. Untersuchung und Entwicklung von Grammatiken regulärer Sprachen</p> <p>(a) Erarbeitung der formalen Darstellung regulärer Grammatiken</p> <p>(b) Untersuchung, Modifikation und Entwicklung von Grammatiken</p> <p>(c) Entwicklung von endlichen Automaten zum Erkennen regulärer Sprachen die durch Grammatiken gegeben werden</p>	<ul style="list-style-type: none"> ◆ 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 	<p>Beispiele:</p> <ul style="list-style-type: none"> ◆ reguläre Grammatik für Wörter mit ungerader Parität, ◆ Grammatik für Wörter, die bestimmte Zahlen repräsentieren, ◆ Satzgliederungsgrammatik <p>Materialien: (s.o.)</p>

(d) Entwicklung regulärer Grammatiken zu endlichen Automaten	Graphen dar und überführen sie in die jeweils andere Darstellungsform (D),	
3. Grenzen endlicher Automaten	<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>Beispiel:</p> <ul style="list-style-type: none"> ◆ Klammersausdrücke, $a^n b^n$ im Vergleich zu $(ab)^n$

Unterrichtsvorhaben Q2-III

Thema: Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

Leitfragen: Wie sind die strukturellen Hauptbestandteile eines Computers und wie kann man sich die Ausführung eines maschinennahen 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: ca. 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, 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 Befehlen 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>Beispiel:</p> <ul style="list-style-type: none"> ◆ Addition von 4 zu einer eingegeben Zahl mit einem Rechnermodell <p>Materialien:</p> <ul style="list-style-type: none"> ◆ Zusatzmaterial zum Kernlehrplan, Unterrichtsvorhaben Q2-III.1 http://www.schulentwicklung.nrw.de/materialdatenbank/nutzersicht/materialeintrag.php?matId=4354 (Stand 06.09.2015)
<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:</p> <ul style="list-style-type: none"> ◆ Halteproblem <p>Zusatzmaterial zum Kernlehrplan, Unterrichtsvorhaben Q2-III.2 http://www.schulentwicklung.nrw.de/materialdatenbank/nutzersicht/materialeintrag.php?matId=4354 (Stand 06.09.2015)</p>

Unterrichtsvorhaben Q2-IV

Wiederholung und Vertiefung ausgewählter Kompetenzen und Inhalte des ersten Jahrs der Qualifikationsphase

2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit.

In Absprache mit der Lehrerkonferenz sowie unter Berücksichtigung des Schulprogramms hat die Fachkonferenz Informatik der Heinrich-Böll--Gesamtschule 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 Schülerinnen und Schüler.
3. Die Unterrichtsgestaltung ist auf die Ziele und Inhalte abgestimmt.
4. Medien und Arbeitsmittel sind schülernah gewählt.
5. Die Schülerinnen und Schüler erreichen einen Lernzuwachs.
6. Der Unterricht fördert eine aktive Teilnahme der Schülerinnen und Schüler.
7. Der Unterricht fördert die Zusammenarbeit zwischen den Schülerinnen und Schülern und bietet ihnen Möglichkeiten zu eigenen Lösungen.
8. Der Unterricht berücksichtigt die individuellen Lernwege der einzelnen Schülerinnen und Schüler.
9. Die Schülerinnen und Schüler 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 Leistungsbewertung und Leistungsrückmeldung

Auf der Grundlage von §13 - §16 der APO-GOST sowie Kapitel 3 des Kernlehrplans Informatik für die gymnasiale Oberstufe hat die Fachkonferenz der Heinrich-Böll-Gesamtschule im Einklang mit dem entsprechenden schulbezogenen Konzept die nachfolgenden Grundsätze zur Leistungsbewertung und Leistungsrückmeldung beschlossen. Die nachfolgenden Absprachen stellen die Minimalanforderungen an das lerngruppenübergreifende gemeinsame Handeln der Fachgruppenmitglieder dar. Bezogen auf die einzelne Lerngruppe kommen ergänzend weitere der in den Folgeabschnitten genannten Instrumente der Leistungsüberprüfung zum Einsatz.

2.3.1 Beurteilungsbereich Klausuren

Verbindliche Absprachen:

Bei der Formulierung von Aufgaben werden die für die Abiturprüfungen geltenden Operatoren des Faches Informatik schrittweise eingeführt, erläutert und dann im Rahmen der Aufgabenstellungen für die Klausuren benutzt.

Instrumente:

- Einführungsphase: 1 Klausur je Halbjahr
Dauer der Klausur: 2 Unterrichtsstunden
- Grundkurse Q 1: 2 Klausuren je Halbjahr
Dauer der Klausuren: 2 Unterrichtsstunden
- Grundkurse Q 2.1: 2 Klausuren
Dauer der Klausuren: 3 Unterrichtsstunden
- Grundkurse Q 2.2: 1 Klausur unter Abiturbedingungen
- Anstelle einer Klausur kann gemäß dem Beschluss der Lehrerkonferenz in Q 1.1 eine Facharbeit geschrieben werden.

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

Kriterien:

Die Bewertung der schriftlichen Leistungen in Klausuren orientiert sich an dem zu formulierenden Erwartungshorizont. Das Ziel der Formulierung des Erwartungshorizonts ist in der unter Abiturbedingungen geschriebenen Klausur das Zuordnungsschema der Notenstufen des schriftlichen Zentralabiturs.

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

Die Note ausreichend (5 Punkte) soll bei Erreichen von 45% des Erwartungshorizonts erteilt werden.

2.3.2 Beurteilungsbereich Sonstige Mitarbeit

Den Schülerinnen und Schülern werden die Kriterien zum Beurteilungsbereich „sonstige Mitarbeit“ zu Beginn des Schuljahres genannt.

Verbindliche Absprachen der Fachkonferenz

Die Fachkonferenz Informatik legt fest, dass in der Einführungsphase und in der Qualifikationsphase 1 und 2 projektorientiertes Arbeiten erlernt und durchgeführt wird.

Leistungsaspekte

Mündliche Leistungen

- Beteiligung am Unterrichtsgespräch
- Zusammenfassung zur Vor- und Nachbereitung des Unterrichts
- Präsentation von Arbeitsergebnissen
- Referate
- Mitarbeit in Partner-/Gruppenarbeitsphasen

Praktische Leistungen am Computer

- Implementierung, Test und Anwendung von Informatiksystemen

Sonstige schriftliche Leistungen

- Arbeitsmappe und Arbeitstagebuch zu einem durchgeführten Unterrichtsvorhaben
- Lernerfolgsüberprüfung durch kurze schriftliche Übungen
Schriftliche Übungen können bis zu 2 mal pro Halbjahr im Umfang von ca. 20 Minuten stattfinden
- Bearbeitung von schriftlichen Aufgaben im Unterricht

Kriterien

Die folgenden allgemeinen Kriterien gelten sowohl für die mündlichen als auch für die schriftlichen Formen der sonstigen Mitarbeit.

Die Bewertungskriterien stützen sich auf

- die Qualität der Beiträge,
- die Quantität der Beiträge und
- die Kontinuität der Beiträge.

Besonderes Augenmerk ist dabei auf

- die sachliche Richtigkeit,
- die angemessene Verwendung der Fachsprache,

- die Darstellungskompetenz,
- die Komplexität und den Grad der Abstraktion,
- die Selbstständigkeit im Arbeitsprozess,
- die Präzision und
- die Differenziertheit der Reflexion zu legen.

Bei Gruppenarbeiten auch auf

- das Einbringen in die Arbeit der Gruppe,
- die Durchführung fachlicher Arbeitsanteile und
- die Qualität des entwickelten Produktes.

Bei Projektarbeit darüber hinaus auf

- die Dokumentation des Arbeitsprozesses,
- den Grad der Selbstständigkeit,
- die Reflexion des eigenen Handelns und
- die Aufnahme von Beratung durch die Lehrkraft.

Grundsätze der Leistungsbewertung und Beratung

Die Grundsätze der Leistungsbewertung werden zu Beginn eines jeden Schuljahres den Schülerinnen und Schülern transparent gemacht. Leistungsrückmeldungen können erfolgen

- nach einer mündlichen Überprüfung,
- bei Rückgabe von schriftlichen Leistungsüberprüfungen,
- nach Abschluss eines Projektes,
- nach einem Vortrag oder einer Präsentation,
- bei auffälligen Leistungsveränderungen,
- auf Anfrage,
- als Quartalsfeedback und
- zu Eltern- oder Schülersprechtagen.

Die Leistungsrückmeldung kann

- durch ein Gespräch mit der Schülerin oder dem Schüler,
- durch einen Feedbackbogen,
- durch die schriftliche Begründung einer Note oder
- durch eine individuelle Lern-/Förderempfehlung

erfolgen.

Leistungsrückmeldungen erfolgen auch in der Einführungsphase im Rahmen der kollektiven und individuellen Beratung zur Wahl des Faches Informatik als fortgesetztes Grund- oder Leistungskursfach in der Qualifikationsphase.

Kriterien für die Bewertung der Sonstigen Mitarbeit

Die Schülerinnen und Schüler erhalten zu Beginn eines Schuljahres eine konkretisierte Form der Bewertungskriterien für die Sonstige Mitarbeit. Neben den allgemeinen Kriterien werden Kompetenzerwartungen in allen Kernkompetenzen formuliert.

Die kompetenzorientierten Kriterien sind von Unterrichtseinheit zu Unterrichtseinheit verschieden gewichtet und können teilweise völlig entfallen. In einer Unterrichtseinheit zur theoretischen Informatik entfällt in natürlicher Weise die Kernkompetenz „Implementieren“. Daher ist es erforderlich, vor jedem Unterrichtsvorhaben die Schülerinnen und Schüler über solche Besonderheiten zu informieren. Daher ist es nicht notwendig, zu jedem Vorhaben eine eigene Kriterienliste zu erstellen.

Im Kapitel 2.1.2 zu den konkretisierten Unterrichtsvorhaben werden zahlreiche Abstufungen und Spezifizierungen der Kernkompetenzen formuliert. Auch darüber sollen die Schülerinnen und Schüler vor jedem Unterrichtsvorhaben informiert werden. Jede der formulierten Kompetenzen lassen sich aber eindeutig einer Kernkompetenz zuordnen. Daher ist eine Anpassung der Kriterienliste für das einzelne Unterrichtsvorhaben nicht erforderlich.

Der Kernlehrplan fordert eine eindeutig feststellbare Progression der Kompetenzen. Diese ist in der Formulierung der Kompetenzerwartungen bis zum Ende der Einführungsphase, bzw. bis zum Ende der Qualifikationsphase erkennbar. Teilweise handelt es sich um Formulierungen, die sich nur in einem Wort unterscheiden, aber sehr bedeutsam sind. Daneben wird die Progression auch durch die Nennung von Kompetenzen in der Qualifikationsphase erkennbar, die in der Einführungsphase nicht gefordert werden.

Daher hat der Fachbereich Informatik einen Kriterienbogen erstellt, der auf den beiden Folgeseiten wiedergegeben wird.

Die Kompetenzen sind nach steigendem Anspruchsniveau angeordnet:

1. Argumentieren (A)
2. Modellieren (M)
3. Implementieren (I)
4. Darstellen und Interpretieren (D)
5. Kommunizieren und Kooperieren (K)

Kriterien für die Bewertung der Sonstigen Mitarbeit im Fach Informatik

Allgemeine Kriterien

Ich ...	immer	oft	manchmal	Seiten oder nie
halte mich an die Grundregeln des Unterrichts. (Z.B. Pünktlichkeit, Vollständigkeit des Materials, Konzentration auf den Unterricht)				
nehme am Unterricht aktiv teil. (Durch selbständiges, konzentriertes und zügiges Arbeiten)				
kann verantwortungsvoll in der Gruppe arbeiten. (In allen Arbeitsformen wie Modellieren, Implementieren, Präsentieren, usw.)				
halte die Gesprächsregeln ein. (Wie Respekt und Fairness in der Sprache und im Umgang mit den Beiträgen anderer)				
bereite den Unterricht vor und nach. (Z.B. durch die Erledigung der Hausaufgaben, das Lernen der Fachsprache, usw.)				

Kompetenzerwartungen in der Einführungs- und Qualifikationsphase

Von den allgemeinen Anforderungen abweichende Erwartungen der jeweiligen Phase sind eingerückt und entsprechend gekennzeichnet.

Ich kann ...	immer	oft	weniger	kaum bis gar nicht
Argumentieren				
methodische Vorgehensweisen erläutern und begründen. Ich bin in der Lage, Entwurfs- und Implementationsentscheidungen sowie Aussagen über Informatiksysteme zu treffen.				
informatische Modelle analysieren und erläutern.				
Computerprogramme analysieren und erläutern.				
EF [die Angemessenheit informatischer Modelle beurteilen.]				
Q 1/2 [die Angemessenheit von Modellierungen und Implementationen beurteilen.]				
Q 1/2 [im Problemlösungsprozess Alternativen aufzeigen und Auswahlentscheidungen begründen.]				
Q 1/2 [informatische Modelle und Informatiksysteme hinsichtlich ihrer Möglichkeiten, Grenzen und Auswirkungen erläutern und beurteilen.]				
Modellieren				
informatische Modelle zu kontextbezogenen Problemstellungen konstruieren.				
informatische Modelle modifizieren und erweitern.				
Q 1/2 [im Modellierungsprozess geeignete Lösungsstrategien anwenden.]				

Ich kann ...	immer	oft	weniger	kaum bis gar nicht
Implementieren				
auf der Grundlage von Modellen oder Modellausschnitten Computerprogramme implementieren.				
Computerprogramme modifizieren und erweitern.				
Computerprogramme testen und korrigieren.				
Darstellen und Interpretieren				
Daten interpretieren sowie Beziehungen und Abläufe, die in Form von textuellen und grafischen [oder formalen] Darstellungen gegeben sind, erläutern.				
gegebene textuelle und grafische [oder formale] Darstellungen informatischer Zusammenhänge in die jeweils andere Darstellungsform überführen.				
informatische Modelle in Texten, Tabellen, Diagrammen [Formalismen] und Grafiken darstellen.				
Kommunizieren und Kooperieren				
Fachausdrücke bzw. Fachsprache bei der Kommunikation über informatische Sachverhalte verwenden.				
in Gruppen- und Partnerarbeit kommunizieren und kooperieren.				
Arbeitsabläufe und Ergebnisse präsentieren.				
Q 1/2 [kooperatives und eigenverantwortliches Arbeiten organisieren und koordinieren.]				
Q 1/2 [den Arbeitsprozess strukturieren, Schnittstellen vereinbaren und Ergebnisse zusammenführen.]				
Q 1/2 [Arbeitsorganisation, Arbeitsabläufe und Ergebnisse beurteilen.]				

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.

Vorbereitung auf die Erstellung der Facharbeit

Möglichst schon 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.

4 Qualitätssicherung und Evaluation

Das schulinterne Curriculum stellt keine starre Größe dar, sondern ist als „lebendes Dokument“ zu betrachten. Dementsprechend sind die Inhalte stetig zu überprüfen, um ggf. Modifikationen vornehmen zu können. Die Fachkonferenz (als professionelle Lerngemeinschaft) trägt durch diesen Prozess zur Qualitätsentwicklung und damit zur Qualitätssicherung des Faches bei.

Durch Diskussion der Aufgabenstellung von Klausuren in Fachdienstbesprechungen und eine regelmäßige Erörterung der Ergebnisse von Leistungsüberprüfungen wird ein hohes Maß an fachlicher Qualitätssicherung erreicht. Das schulinterne Curriculum (siehe 2.1) ist zunächst bis 2017 für den ersten Durchgang durch die gymnasiale Oberstufe nach Erlass des Kernlehrplanes verbindlich. Erstmals nach Ende der Einführungsphase im Sommer 2015 werden in einer Sitzung der Fachkonferenz Erfahrungen ausgetauscht und ggf. Änderungen für den nächsten Durchgang der Einführungsphase beschlossen, um erkannten ungünstigen Entscheidungen schnellstmöglich entgegenwirken zu können.

Nach Abschluss des Abiturs 2017 wird die Fachkonferenz Informatik auf der Grundlage ihrer Unterrichtserfahrungen eine Gesamtsicht des schulinternen Curriculums vornehmen und ggf. eine Beschlussvorlage für die erste Fachkonferenz des folgenden Schuljahres erstellen.