

Schulinterner Lehrplan zum Kernlehrplan für die gymnasiale Oberstufe

Informatik

(Stand: 13.05.2015)

Inhalt

1	Die Fachgruppe Informatik des Gymnasiums der Stadt Lage	2
2	Entscheidungen zum Unterricht	3
2.1	Unterrichtsvorhaben	3
2.1.1	Konkretisierte Unterrichtsvorhaben	4
2.2	Grundsätze der fachmethodischen und fachdidaktischen Arbeit	37
2.3	Grundsätze der Leistungsbewertung und Leistungsrückmeldung	38
2.3.1	Beurteilungsbereich Klausuren	38
2.3.2	Beurteilungsbereich Sonstige Mitarbeit	39
3	Entscheidungen zu fach- und unterrichtsübergreifenden Fragen	40
4	Qualitätssicherung und Evaluation	40

1 Die Fachgruppe Informatik des Gymnasiums der Stadt Lage

Das Fach Informatik wird am Gymnasium der Stadt Lage ab der Jahrgangsstufe 8 als Fachkombination mit Physik im Wahlpflichtbereich II (WP II) zweistündig unterrichtet. In der zweijährigen Laufzeit dieser Kurse wird in altersstufengerechter Weise unter anderem auf Grundlagen der Algorithmik am Beispiel didaktischer Lernumgebungen und anwendungsbezogener Informatik am Beispiel elektronischer Schaltungen eingegangen.

In der Sekundarstufe II bietet das Gymnasium der Stadt Lage Kurse im Fach 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 Hilfe der Programmiersprache Java durchgeführt.

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

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

Zurzeit besteht die Fachschaft Informatik des Gymnasiums der Stadt Lage aus zwei Lehrkräften, denen drei Computerräume mit je 14 Computerarbeitsplätzen zur Verfügung stehen. Alle Arbeitsplätze sind an das schulinterne Rechnernetz angeschlossen, so dass Schülerinnen und Schüler Zugang zum zentralen Server der Schule zum Zugriff auf ihre eigenen Daten, zur Recherche im Internet oder zur Bearbeitung schulischer Aufgaben haben.

Im Differenzierungsbereich der Jahrgänge 8 und 9 ist der 90-minütige Unterricht vom Daltonkonzept entkoppelt. Von den 135 Minuten Unterrichtszeit in den Grundkursen der SII entfallen ein Drittel auf die Daltonarbeitszeit.

2 Entscheidungen zum Unterricht

2.1 Unterrichtsvorhaben

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

Der ausgewiesene Zeitbedarf versteht sich als grobe Orientierungsgröße, die nach Bedarf über- oder unterschritten werden kann. Um Freiraum für Vertiefungen, besondere Schülerinteressen, aktuelle Themen bzw. die Erfordernisse anderer besonderer Ereignisse (z.B. Praktika, Kursfahrten o.ä.) zu erhalten, wurden im Rahmen dieses schulinternen Lehrplans ca. 80 Prozent der Bruttounterrichtszeit verplant.

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 Konkretisierte Unterrichtsvorhaben

Unterrichtsvorhaben EF-I

Thema: Geschichte der digitalen Datenverarbeitung bis zur Von-Neumann-Architektur und Einführung in die Grundlagen, Anwendungsgebiete und Verarbeitung binärer Codierung

Leitfragen: *Welche Entwicklung durchlief die moderne Datenverarbeitung? Wie werden binäre Informationen gespeichert und wie können sie davon ausgehend weiter verarbeitet werden? Wie ist der Grundaufbau einer digitalen Rechenmaschine?*

Vorhabenbezogene Konkretisierung:

Schülerinnen und Schüler sollen zunächst selbstständig informatische Themenbereiche aus dem Kontext der Geschichte der Datenverarbeitung erarbeiten. Diese Themenbereiche werden in Kleingruppen bearbeitet und in Form von Präsentationen vorgestellt. Schülerinnen und Schüler sollen dabei mit Unterstützung des Lehrenden selbstständige Recherchen zu ihren Themen anstellen und auch eine sinnvolle Eingrenzung ihres Themas vornehmen.

Im Anschluss daran werden die binäre Speicherung und Verarbeitung thematisiert. Nach einer ersten grundlegenden Einordnung des digitalen Prinzips überführen die Schülerinnen und Schüler Dezimalzahlen in das binäre Zahlensystem und umgekehrt.

Zum Abschluss soll der grundlegende Aufbau eines Rechnersystems im Sinne der von-Neumann-Architektur erarbeitet werden.

Zeitbedarf: ca. 9 Stunden

Sequenzierung des Unterrichtsvorhabens: nächste Seite

Unterrichtsvorhaben EF I: Geschichte der digitalen Datenverarbeitung bis zur Von-Neumann-Architektur und Einführung in die Grundlagen, Anwendungsgebiete und Verarbeitung binärer Codierung Leitfragen: Welche Entwicklung durchlief die moderne Datenverarbeitung? Wie werden binäre Informationen gespeichert und wie können sie davon ausgehend weiter verarbeitet werden? Wie ist der Grundaufbau einer digitalen Rechenmaschine?		Zeitbedarf: 9 Stunden
Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien, Methoden
1. Selbstständige Erarbeitung von Themen durch die Schülerinnen und Schüler (a) Themen zur Erarbeitung in Kleingruppen (b) Vorstellung und Diskussion durch Schülerinnen und Schüler	Die Schülerinnen und Schüler <ul style="list-style-type: none"> • bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A) • erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A) • stellen ganze Zahlen und Zeichen in Binärcodes dar (D) • interpretieren Binärcodes als Zahlen und Zeichen (D) • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K) 	<i>Mögliche Themen:</i> <ul style="list-style-type: none"> • „Eine kleine Geschichte der Digitalisierung: vom Morsen zum modernen Digitalcomputer“ • „Von Nullen, Einsen und mehr: Stellenwertsysteme und wie man mit ihnen rechnet“ • „Kodieren von Texten und Bildern: ASCII, RGB und mehr“ • „Technische Grundlagen des Internets“ • „Auswirkungen der Digitalisierung und des Einsatzes von Informationssystemen auf die Gesellschaft“
2. Aufbau informatischer Systeme (a) Identifikation des EVA-Prinzips als grundlegende Arbeitsweise in-formatischer Systemen (b) Nachvollziehen der Von-Neumann-Architektur als relevantes Modell der Umsetzung des EVA-Prinzips	<ul style="list-style-type: none"> • beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A) 	

Unterrichtsvorhaben EF-II

Thema: Grundlagen der objektorientierten Analyse, Modellierung und Implementierung

Leitfragen: *Wie lassen sich Gegenstandsbereiche informatisch modellieren und realisieren?*

Vorhabenbezogene Konkretisierung:

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

Das Ziel dieses Unterrichtsvorhabens besteht darin, das Verhalten von Objekten flexibel zu programmieren. Ein erster Schwerpunkt liegt dabei auf der Erarbeitung von Kontrollstrukturen. Die Strukturen Wiederholung und bedingte Anweisung werden an einfachen Beispielen eingeführt und anschließend anhand komplexerer Problemstellungen erprobt. Da die zu entwickelnden Algorithmen zunehmend umfangreicher werden, werden systematische Vorgehensweisen zur Entwicklung von Algorithmen thematisiert.

Ein zweiter Schwerpunkt des Unterrichtsvorhabens liegt auf dem Einsatz von Variablen. Beginnend mit lokalen Variablen, die in Methoden und Zählschleifen zum Einsatz kommen, über Variablen in Form von Parametern und Rückgabewerten von Methoden, bis hin zu Variablen, die die Attribute einer Klasse realisieren, lernen die Schülerinnen und Schüler die unterschiedlichen Einsatzmöglichkeiten des Variablenkonzepts anzuwenden.

Anschließend können die wesentlichen Eigenschaften von Algorithmen wie z.B. Korrektheit, Terminiertheit, Effizienz und Verständlichkeit sowie die Schritte einer Algorithmenentwicklung erarbeitet werden (Klärung der Anforderung, Visualisierung, Zerlegung in Teilprobleme).

Zeitbedarf: ca. 33 Stunden

Sequenzierung des Unterrichtsvorhabens: nächste Seite

Unterrichtsvorhaben EF II: Grundlage der objektorientierten Analyse, Modellierung und Implementierung		Zeitbedarf: 33 Stunden
Leitfrage: Wie lassen sich Gegenstandsbereiche informatisch modellieren und realisieren?		
Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien, Methoden
1. Einführung in objektorientiertes Denken und Modellieren (a) Klassen (b) Objekte (c) Methoden (d) Diagramme	Die Schülerinnen und Schüler ■ ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften und ihre Operationen (M) ■ analysieren und erläutern einfache Algorithmen und Programme (A) ■ modifizieren einfache Algorithmen und Programme (I) ■ nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D) ■ implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I) ■ entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M) ■ testen Programme schrittweise anhand von Beispielen (I) ■ ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen zu (M) ■ dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D) ■ implementieren Klassen in einer	<i>Beispiel:</i> Textarbeit (Substantiv-Verb-Methode), Kundendatei eines Autohauses, etc. <i>Material:</i> IO-Stick von Tino Hempel <i>Beispiel:</i> Schatzräuber-Projekt in Greenfoot <i>Möglich:</i> Partnerpuzzle zum Thema bedingte Anweisungen und log. Verknüpfungen Mögliche Wiederholung und Vertiefung mit Roboter-Szenario nach den Herbstferien Mögliche Erweiterung des Schatzräuberszenarios
2. Einführung in die objektorientierte Programmierung mit Java (a) Klassen (b) Objekte (c) Methoden (d) Anweisungen (e) Stylekonventionen	(continued from previous row)	(continued from previous row)
3. Kontrollstrukturen und Algorithmenentwurf (a) Bedingte Anweisungen (b) Logische Verknüpfungen (c) While- und For-Schleifen (d) Variablen (int) (e) Struktogramme	(continued from previous row)	(continued from previous row)

<p>4. Methoden und Variablen</p> <ul style="list-style-type: none"> (a) Aufbau von Methoden (b) Konstruktor (c) Variablen (d) Datentypen 	<p>Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I)</p> <p>■ implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I)</p>	<p><i>Beispiel:</i> Roboter-Szenario in Greenfoot</p> <p><i>Möglich:</i> Projekt Kamelrennen (Spiel)</p>
<p>5. Algorithmus</p> <ul style="list-style-type: none"> (a) Eigenschaften von Algorithmen (b) Algorithmen aus Alltagsbeispielen 		<p><i>Beispiel:</i> Formale Darstellung alltäglicher Vorgänge</p>

Unterrichtsvorhaben EF-III

Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen und Sichtbarkeitsbereiche von Attributen und Methoden

Leitfragen: *Wie werden realistische Systeme anforderungsspezifisch reduziert, als Entwurf modelliert und implementiert?*

Vorhabenbezogene Konkretisierung:

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

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

Nach diesem ersten Modellierungsschritt werden über Klassendokumentationen Implementationsdiagramme entwickelt. Danach werden die Implementationsdiagramme in Java programmiert.

In diesem Unterrichtsvorhaben werden die grundlegenden Begriffe der Objektorientierung und Modellierungswerkzeuge wie Objektdiagramme und Klassendiagramme eingeführt.

Zeitbedarf: ca. 27 Stunden

Sequenzierung des Unterrichtsvorhabens: nächste Seite

Unterrichtsvorhaben EF III: Modellierung und Implementierung von Klassen- und Objektbeziehungen und Sichtbarkeitsbereiche von Attributen und Methoden		Zeitbedarf: 27 Stunden
Leitfrage: Wie werden realistische Systeme anforderungsspezifisch reduziert, als Entwurf modelliert und implementiert?		
Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien, Methoden
1. Unified Modeling Language (UML-Diagramme) (a) Klassen und Objekte (b) Entwurfsdiagramm (c) Implementationsdiagramm (d) Objektdiagramm	Die Schülerinnen und Schüler <ul style="list-style-type: none"> ■ ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften und ihre Operationen (M) ■ nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D) ■ interpretieren Fehlermeldungen und korrigieren den Quellcode (I) ■ ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen zu (M) 	Mögliche Überleitung von Greenfoot zu BlueJ Mögliches Programm für Diagramme: DIA
2. Datenkapselung (a) Sichtbarkeitsbereiche (public, private) (b) get- und set-Methoden	<ul style="list-style-type: none"> ■ modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M) ■ ■ analysieren und erläutern eine objektorientierte Modellierung (A) ■ ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M) ■ stellen den Zustand eines Objekts dar (D) ■ ■ stellen Klassen- und Assoziationsbeziehungen in Diagrammen grafisch dar (D) 	<i>Beispielprogramm:</i> Einfaches Spiel z.B. Verflixte 7 in BlueJ

Unterrichtsvorhaben EF-IV

Thema: Such- und Sortieralgorithmen

Leitfragen: *Wie können Objekte bzw. Daten effizient gesucht und sortiert werden?*

Vorhabenbezogene Konkretisierung:

Zunächst lernen die Schülerinnen und Schüler Arrays als eine erste Datenstruktur kennen.

Im zweiten Teil dieses Unterrichtsvorhaben wird die Datenstruktur Array erneut bei der Erarbeitung von Such- und Sortieralgorithmen aufgegriffen. Der Schwerpunkt des Vorhabens liegt dabei auf den Algorithmen selbst und nicht auf deren Implementierung in einer Programmiersprache, auf die in diesem Vorhaben vollständig verzichtet werden soll.

Daran anschließend lernen die Schülerinnen und Schüler Strategien des Sortierens (z.B.: Selection Sort, Insertion Sort, Bubble Sort, Quick Sort) kennen. Der Einstieg über die intuitiven Sortierstrategien dient dazu, die jeweiligen Strategien handlungsorientiert zu erkunden und intuitive Effizienzbetrachtungen der Suchalgorithmen vorzunehmen.

Schließlich wird die Effizienz unterschiedlicher Sortierverfahren beurteilt.

Durch die Analyse von verschiedenen Fehlermeldungen des Java-Compilers werden die Schülerinnen und Schüler in der selbstständigen Fehlerkorrektur geschult.

Zeitbedarf: ca. 18 Stunden

Sequenzierung des Unterrichtsvorhabens: nächste Seite

Unterrichtsvorhaben EF IV: Such- und Sortieralgorithmen		Zeitbedarf: 18 Stunden
Leitfrage: Wie können Objekte bzw. Daten effizient gesucht und sortiert werden?		
Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien, Methoden
1. Statische Datenstruktur Array (a) Eigenschaften (b) Operationen auf Arrays (c) Arrays mit komplexen Datentypen	Die Schülerinnen und Schüler ■ nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D) ■ implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I) ■ ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M) ■ analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D) ■ entwerfen einen weiteren Algorithmus zum Sortieren (M) ■ beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeitaufwand und Speicherplatzbedarf ■ interpretieren Fehlermeldungen und korrigieren den Quellcode (I) ■ nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K)	<i>Beispiel:</i> Bundesliga-Tabelle in BlueJ <i>Möglich:</i> Gruppenpuzzle zu Sortierverfahren Empfohlene Datenstruktur: Array <i>Möglich:</i> Glossar-Erstellung in Gruppen zu allen Begrifflichkeiten und Fachkonzepten, Youtube Video: „Hungarian Dance“
2. Suchen und Sortieren (a) Intuitive Sortierstrategien (b) Sortierverfahren (Insertion-, Selection-, Bubble- und Quick Sort) (c) Laufzeitanalyse		
3. Fehlermeldungen in Java (a) Analyse häufiger Fehlermeldungen in Java (b) Fehlerkorrektur im Quelltext		

Unterrichtsvorhaben EF-V

Thema: Das Konzept der Vererbung in Java

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

Vorhabenbezogene Konkretisierung:

In diesem Unterrichtsvorhaben wird das Prinzip der Vererbung im objektorientierten Sinne angesprochen. Dazu werden die wichtigsten Varianten der Vererbung anhand eines Beispielprojekts erarbeitet. Zunächst wird die Vererbung als Spezialisierung im Sinne einer einfachen Erweiterung einer Oberklasse vorgestellt. Daraufhin wird das Verständnis von Vererbung um den Aspekt der späten Bindung erweitert, indem Methoden einer Oberklasse überschrieben werden. Modellierungen sollen in Form von Implementationsdiagrammen erstellt werden.

Zum Abschluss kann auf das Prinzip der abstrakten Klassen und Methoden eingegangen werden. Dieser Inhalt ist aber nicht obligatorisch für die Einführungsphase.

Zeitbedarf: ca. 6 Stunden

Sequenzierung des Unterrichtsvorhabens: nächste Seite

Unterrichtsvorhaben EF V: Das Konzept der Vererbung in Java		Zeitbedarf: 6 Stunden
Leitfrage: Wie lassen sich komplexere Beziehungen zwischen Objekten und Klassen realisieren?		
Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien, Methoden
<p>1. Vererbung</p> <ul style="list-style-type: none"> (a) Vererbungshierarchie (Ober- und Unterklasse) (b) Implementierung in Java (c) Polymorphie (Überschreiben von Methoden) (d) Instanceof-Abfrage → Typecast (e) Abstrakte Klassen und Methoden 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ■ modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M) ■ ■ modellieren Klassen unter Verwendung von Vererbung (M) ■ ■ stellen die Kommunikation zwischen Objekten grafisch dar (M) ■ stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D) ■ implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I) ■ ■ interpretieren Fehlermeldungen und korrigieren den Quellcode (I) ■ analysieren und erläutern eine objektorientierte Modellierung (A) 	

Unterrichtsvorhaben EF-VI

Thema: Grundlegende Aspekte der Datensicherheit und Verschlüsselung

Leitfragen: *Wie wichtig ist Datensicherheit in der heutigen Zeit? Wie kann man diese für persönliche Daten sicherstellen?*

Vorhabenbezogene Konkretisierung:

In diesem Unterrichtsvorhaben wird auf grundlegende Aspekte der Datensicherheit im Kontext der informationstechnologischen Gesellschaft eingegangen und darüber diskutiert. Im weiteren Verlauf werden grundlegende Methoden der Kryptographie angesprochen und diese bis zur Idee der asymmetrischen Verschlüsselung (z.B.: RSA-Verschlüsselung) geführt.

Zeitbedarf: ca. 12 Stunden

Sequenzierung des Unterrichtsvorhabens: nächste Seite

Unterrichtsvorhaben EF VI: Grundlegende Aspekte der Datensicherheit und Verschlüsselung		Zeitbedarf: 12 Stunden
Leitfragen: Wie wichtig ist Datensicherheit in der heutigen Zeit? Wie kann man diese für persönliche Daten sicherstellen?		
Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien, Methoden
<p>1. Datensicherheit</p> <p>(a) Datensicherheit, Persönlichkeitsrechte</p> <p>(b) Konflikt persönliche Freiheit vs. Sicherheit</p> <p>2. Kryptographie</p> <p>(a) Substitutionsverfahren: Skytale, Cäsar, Vigènere</p> <p>(b) Asymmetrische Verschlüsselungsverfahren</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • Diskutieren über Aspekte (A) der Datensicherheit und stellen ihre Position dar (K) <p>■ erläutern und begründen Funktionsweise und Eignung verschiedener Verschlüsselungsverfahren (A)</p>	<p><i>Material:</i> Youtube-Video Mathematik zum Anfassen: Kryptographie, Cäsar-Scheibe, Vigènere-Quadrat, etc.</p>

Unterrichtsvorhaben

Thema: Geschichte der digitalen Datenverarbeitung bis zur Von-Neumann-Architektur und Einführung in die Grundlagen, Anwendungsgebiete und Verarbeitung binärer Codierung

Leitfragen: *Welche Entwicklung durchlief die moderne Datenverarbeitung? Wie werden binäre Informationen gespeichert und wie können sie davon ausgehend weiter verarbeitet werden? Wie ist der Grundaufbau einer digitalen Rechenmaschine?*

Vorhabenbezogene Konkretisierung:

Schülerinnen und Schüler sollen zunächst selbstständig informatische Themenbereiche aus dem Kontext der Geschichte der Datenverarbeitung erarbeiten. Diese Themenbereiche werden in Kleingruppen bearbeitet und in Form von Präsentationen vorgestellt. Schülerinnen und Schüler sollen dabei mit Unterstützung des Lehrenden selbstständige Recherchen zu ihren Themen anstellen und auch eine sinnvolle Eingrenzung ihres Themas vornehmen.

Im Anschluss daran werden die binäre Speicherung und Verarbeitung thematisiert. Nach einer ersten grundlegenden Einordnung des digitalen Prinzips überführen die Schülerinnen und Schüler Dezimalzahlen in das binäre Zahlensystem und umgekehrt.

Zum Abschluss soll der grundlegende Aufbau eines Rechnersystems im Sinne der von-Neumann-Architektur erarbeitet werden.

Zeitbedarf: ca. 9 Stunden

Sequenzierung des Unterrichtsvorhabens: nächste Seite

Unterrichtsvorhaben Q1-I:

Thema: Wiederholung der objektorientierten Modellierung und Programmierung

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

Vorhabenbezogenen Konkretisierung:

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

Die Schülerinnen und Schülern erläutern und modifizieren den ersten Entwurf und modellieren sowie implementieren weitere Klassen und Methoden für eine entsprechende Anwendung. Klassen und ihre Beziehungen werden in einem Implementationsdiagramm dargestellt. Dabei werden Sichtbarkeitsbereiche zugeordnet. Exemplarisch wird eine Klasse dokumentiert. 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: 8 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Wiederholung und Erweiterung der objektorientierten Modellierung und Programmierung durch Analyse und Erweiterung eines kontextbezogenen Beispiels	Die Schülerinnen und Schüler <ul style="list-style-type: none">analysieren und erläutern objektorientierte Modellierungen (A),	<i>BlueJ</i> (z.B. einfache Projekte mit GLOOP) <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.1-Wiederholung

<p>(a) Analyse der Problemstellung (b) Analyse der Modellierung (Implementationsdiagramm) (c) Erweiterung der Modellierung im Implementationsdiagramm (Vererbung, abstrakte Klasse) (d) Kommunikation zwischen mindestens zwei Objekten (grafische Darstellung) (e) Dokumentation von Klassen (f) Implementierung der Anwendung oder von Teilen der Anwendung</p>	<ul style="list-style-type: none"> • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), • ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen (D), • stellen die Kommunikation zwischen Objekten grafisch dar (D). 	
--	--	--

Unterrichtsvorhaben Q1-II:

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

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

Vorhabenbezogene Konkretisierung:

An Anwendungsbeispielen werden die linearen Datenstrukturen Liste, Schlange und Stapel eingeführt. Hierbei soll mindestens Schlange und/oder Stapel selbst implementiert werden. In mindestens einem weiteren Anwendungskontext wird die Verwaltung von Daten in Schlangen, Stapeln oder Listen vertieft. Modellierungen werden dabei in Entwurfs- und Implementationsdiagrammen dargestellt.

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse <code>Queue</code></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse <code>Queue</code></p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse <code>Queue</code></p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">• erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A),• analysieren und erläutern Algorithmen und Programme (A),• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M),	<p><i>Beispiel:</i> Angemessenes Projekt: Z.B. Wartezimmer eines Arztes, Supermarkt, etc</p>

<p>2. Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse <code>Stack</code></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse <code>Stack</code></p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse <code>Stack</code></p>	<ul style="list-style-type: none"> • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • modifizieren Algorithmen und Programme (I), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen (I), • stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D). 	
<p>3. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse <code>List</code></p> <p>(a) Erarbeitung der Vorteile der Klasse <code>List</code> im Gegensatz zu den bereits bekannten linearen Strukturen</p> <p>(b) Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse <code>List</code>.</p>		
<p>4. Vertiefung - Anwendungen von Listen, Stapeln oder Schlangen in mindestens einem weiteren Kontext</p>		

Unterrichtsvorhaben Q1-III:

Thema: Suchen und Sortieren auf linearen Datenstrukturen

Leitfrage: *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 bzw. wiederholt und implementiert (ebenfalls für lineare Listen und Felder). Hierbei soll auch ein rekursives Sortierverfahren entwickelt werden. Die Implementationen von Quicksort sowie dem Sortieren durch Einfügen werden analysiert und erläutert. Die rekursive Abarbeitung eines Methodenaufrufs von Quicksort wird grafisch dargestellt.

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

Zeitbedarf: 16 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 hinsichtlich ihrer	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),	

<p>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), 	
<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>	<ul style="list-style-type: none"> • 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), 	
<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>	<ul style="list-style-type: none"> • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D). 	

Unterrichtsvorhaben Q1-IV:

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

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

Vorhabenbezogene Konkretisierung:

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

Unter anderem sollen die verschiedenen Baumtraversierungen (Pre-, Post- und Inorder) implementiert werden. Unterschiede bezüglich der Möglichkeit, den Baum anhand der Ausgabe der 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: 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), 	<p><i>Beispiel:</i> Termbaum Der Aufbau von Termen wird mit Hilfe von binären Baumstrukturen verdeutlicht.</p> <p>oder</p> <p><i>Beispiel:</i> Ahnenbaum Die binäre Baumstruktur ergibt sich daraus, dass jede Person genau einen Vater und eine Mutter hat.</p> <p><i>Weitere Beispiele für Anwendungskontexte für binäre Bäume:</i></p> <p><i>Beispiele:</i> Suchbäume, Entscheidungsbäume, Codierungsbäume</p>
<p>2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse <code>BinaryTree</code></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext</p>	<ul style="list-style-type: none"> • verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M), 	

<p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms</p> <p>(c) Erarbeitung der Klasse <code>BinaryTree</code> und beispielhafte Anwendung der Operationen</p> <p>(d) Implementierung der Anwendung oder von Teilen der Anwendung</p> <p>(e) Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf</p>	<ul style="list-style-type: none"> • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • modifizieren Algorithmen und Programme (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen (I), • stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D). 	
<p>3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse <code>BinarySearchTree</code></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramm, grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften</p> <p>(c) Erarbeitung der Klasse <code>BinarySearchTree</code> und Einführung des Interface <code>Item</code> zur Realisierung einer geeigneten Ordnungsrelation</p> <p>(d) Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums</p>		

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

Unterrichtsvorhaben Q1-V:

Thema: Navigieren mit Struktur - Graphen

Leitfragen: *Wie können z.B. (günstige) Wege auf Karten oder in Navigationssystemen ermittelt werden?*

Vorhabenbezogene Konkretisierung:

Anhand von vereinfachten Ortsabstandsgraphen werden Strategien zur Navigation ermittelt.

Zeitbedarf: 6 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Graphen als weiteres Beispiel einer nichtlinearen Datenstruktur</p> <p>(c) Gewichtete und ungewichtete (gerichtete und ungerichtete) Graphen)</p> <p>(d) Suchoperationen auf Graphen (Tiefen- und Breitensuche)</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), • entwickeln iterative und rekursive Algorithmen unter Nutzung der 	<p><i>Beispiel:</i> Ortsabstandskarten, Euler-Wege, etc</p>

	Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M), <ul style="list-style-type: none"> • stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D). 	

Unterrichtsvorhaben Q2-I:

Thema: Sicherheit und Datenschutz in Netzstrukturen

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

Vorhabenbezogene Konkretisierung:

Es werden Sicherheitsaspekte bei Datenspeicherung und Datenübermittlung sowie verschiedene symmetrische und asymmetrische kryptografische Verfahren analysiert und erläutert.

Zeitbedarf: 6 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) Vertraulichkeit, Integrität, Authentizität in Netzwerken sowie	Die Schülerinnen und Schüler <ul style="list-style-type: none"> • analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A), 	<i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben, Verschlüsselung Q1.5 - Zugriff auf Daten in Netzwerken (Download Q1-V.1)

<p>symmetrische und asymmetrische kryptografische Verfahren (Cäsar-, Vigenère-, RSA-Verfahren) als Methoden Daten im Netz verschlüsselt zu übertragen</p>	<ul style="list-style-type: none"> • nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D). 	
		<p><i>Materialien:</i> <i>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1 5 - Datenschutz beim Videocenter, Materialblatt-Datenschutzgesetz</i> (Download Q1-V.2)</p>

Unterrichtsvorhaben Q2-II:

Thema: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten und Struktur von Netzwerken

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

Vorhabenbezogene Konkretisierung:

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

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.

Am Beispiel von Datenbanken wird kontextgebunden der Zugriff aus Netzwerken thematisiert. Dabei werden Grundlagen von Netzwerkprotokollen und das OSI-Schichtenmodell eingeführt sowie die Sicherheit von Daten thematisiert.

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Nutzung von relationalen Datenbanken</p> <p>(a) Aufbau von Datenbanken und Grundbegriffe</p> <ul style="list-style-type: none"> Entwicklung von Fragestellungen zur vorhandenen Datenbank Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, Datenbankschema <p>(b) SQL-Abfragen</p> <ul style="list-style-type: none"> Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL (SELECT (DISTINCT) ...FROM, WHERE, AND, OR, NOT) auf einer Tabelle 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>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A), analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A), analysieren und erläutern eine Datenbankmodellierung (A), erläutern die Eigenschaften normalisierter Datenbankschemata (A), bestimmen Primär- und Sekundärschlüssel (M), ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M), modifizieren eine Datenbankmodellierung (M), modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M), bestimmen Primär- und Sekundärschlüssel (M), überführen Datenbankschemata in vorgegebene Normalformen (M), verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I), ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D), stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D), 	<p><i>Beispiel:</i> VideoCenter</p> <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.</p> <p>http://dokumentation.videocenter.schule.de/old/video/index.html</p> <p>·</p> <p><i>Beispiel:</i> Schulbuchausleihe</p> <p>Unter www.brd.nrw.de/lerntreffs/informatik/structure/material/sek2/datenbank/en.php (abgerufen: 30. 03. 2014) wird eine Datenbank zur Verfügung gestellt, die Daten einer Schulbuch-Ausleihe enthält (über 1000 Entleiher, 200 Bücher mit mehreren tausend Exemplaren und viele Ausleihvorgänge). Die Datenbank kann in OpenOffice eingebunden werden.</p>
<p>2. Modellierung von relationalen Datenbanken</p>		<p><i>Beispiel:</i> Fahrradverleih</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 Datenbankschematas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln <p>(c) Redundanz, Konsistenz und Normalformen</p> <ul style="list-style-type: none"> • Untersuchung einer Datenbank hinsichtlich Konsistenz und Redundanz in einer Anwendungssituation • Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten) 	<ul style="list-style-type: none"> • überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D). • Erläutern Eigenschaften, Funktionsweisen und Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A) 	<p>Der Fahrradverleih <i>BTR (BikesToRent)</i> verleiht unterschiedliche Typen von Fahrrädern diverser Firmen an seine Kunden. Die Kunden sind bei <i>BTR</i> registriert (Name, Adresse, Telefon). <i>BTR</i> kennt von den Fahrradfirmen den Namen und die Telefonnummer. Kunden von <i>BTR</i> können CityBikes, Treckingräder und Mountainbikes ausleihen.</p> <p><i>Beispiele:</i> Reederei, Buchungssystem, Schulverwaltung</p>
<p>3. Struktur von Netzwerken</p>	<ul style="list-style-type: none"> • Beschreiben und erläutern Netzwerktopologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell im Netzwerk (A) 	

Unterrichtsvorhaben Q2-III:

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: 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p>1. Endliche Automaten</p> <p>(a) Vom Automaten in den Schölerinnen und Schölern bekannten Kontexten zur formalen Beschreibung eines endlichen Automaten</p> <p>(b) Untersuchung, Darstellung und Entwicklung endlicher Automaten</p>	<p>Die Schölerinnen und Schöler</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 zur akzeptierten Sprache eines Automaten die zugehörige Grammatik (M), • entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M), • modifizieren Grammatiken regulärer Sprachen (M), • entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M), • stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D), • ermitteln die Sprache, die ein endlicher Automat akzeptiert (D). 	<p><i>Beispiele:</i></p> <p>Cola-Automat, Geldspielautomat, Roboter, Zustandsänderung eines Objekts „Auto“, Akzeptor für bestimmte Zahlen, Akzeptor für Teilwörter in längeren Zeichenketten, Akzeptor für Terme</p> <p><i>Materialien:</i></p> <p>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.2 – Endliche Automaten, Formale Sprachen (Download Q2-II.1)</p>
<p>2. Untersuchung und Entwicklung von Grammatiken regulärer Sprachen</p> <p>(a) Erarbeitung der formalen Darstellung regulärer Grammatiken</p> <p>(b) Untersuchung, Modifikation und Entwicklung von Grammatiken</p> <p>(c) Entwicklung von endlichen Automaten zum Erkennen regulärer Sprachen die durch Grammatiken gegeben werden</p> <p>(d) Entwicklung regulärer Grammatiken zu endlichen Automaten</p>	<p>(This cell is shared with the previous row and contains the same list of competencies.)</p>	<p><i>Beispiele:</i></p> <p>reguläre Grammatik für Wörter mit ungerader Parität, Grammatik für Wörter, die bestimmte Zahlen repräsentieren, Satzgliederungsgrammatik</p> <p><i>Materialien: (s.o.)</i></p>
<p>3. Grenzen endlicher Automaten</p>		

	<ul style="list-style-type: none"> • beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D). 	<i>Beispiele:</i> Klammersausdrücke, $a^n b^n$ im Vergleich zu $(ab)^n$
--	--	--

Unterrichtsvorhaben Q2-IV:

Thema: Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

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

Vorhabenbezogene Konkretisierung:

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

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

Zeitbedarf: 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
1. Von-Neumann-Architektur und die Ausführung maschinennaher Programme	Die Schülerinnen und Schüler	<i>Materialien:</i>

<ul style="list-style-type: none"> a) prinzipieller Aufbau einer von Neumann-Architektur mit CPU, Rechenwerk, Steuerwerk, Register und Hauptspeicher b) einige maschinennahe Befehlen und ihre Repräsentation in einem Binär-Code, der in einem Register gespeichert werden kann c) Analyse und Erläuterung der Funktionsweise eines einfachen maschinennahen Programms 	<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>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 –Von-Neumann-Architektur und maschinennahe Programmierung (Download Q2-III.1)</p>
<p>2. Grenzen der Automatisierbarkeit</p> <ul style="list-style-type: none"> a) Vorstellung des Halteproblems b) Unlösbarkeit des Halteproblems c) Beurteilung des Einsatzes von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen 		<p><i>Beispiel:</i> Halteproblem</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 - Halteproblem (Download Q2-III.2)</p>

Unterrichtsvorhaben Q2-V:

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

2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

Die Fachgruppe Informatik des Gymnasiums der Stadt Lage hat sich auf folgende fachmethodischen und fachdidaktischen Grundsätze geeinigt. In diesem Zusammenhang beziehen sich die Grundsätze 1 bis 14 auf fächerübergreifende Aspekte, die Grundsätze 15 bis 21 sind fachspezifisch angelegt.

Überfachliche Grundsätze:

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

Fachliche Grundsätze:

- 14) Der Unterricht unterliegt der Wissenschaftsorientierung und ist dementsprechend eng verzahnt mit seiner Bezugswissenschaft.
- 15) Der Unterricht ist problemorientiert.
- 16) Der Unterricht folgt dem Prinzip des Exemplarischen und soll ermöglichen, informatische Strukturen und Gesetzmäßigkeiten in den ausgewählten Problemen und Projekten zu erkennen.
- 17) Der Unterricht ist anschaulich sowie gegenwarts- und zukunftsorientiert.
- 18) Der Unterricht ist handlungsorientiert, d.h. projekt- und produktorientiert angelegt.
- 19) Im Unterricht werden sowohl für die Schule didaktisch reduzierte als auch reale Informatiksysteme aus der Wissenschafts-, Berufs- und Lebenswelt eingesetzt.
- 20) 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 Fachgruppe des Gymnasiums der Stadt Lage 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: 3 Klausur im Schuljahr
Dauer der Klausur: 2 Unterrichtsstunden
- Grundkurse Q 1: 2 Klausuren je Halbjahr
Dauer der Klausuren: 2-3 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 1.2 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 erfolgt über ein Raster mit Hilfspunkten, die im Erwartungshorizont den einzelnen Kriterien zugeordnet sind.

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

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

Die Note ausreichend (5 Punkte) soll bei Erreichen von 45 % der Hilfspunkte 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

- Der Schwerpunkt des Unterrichts wird auf die objektorientierte Programmierung in Java gelegt.

Leistungsaspekte

Mündliche Leistungen

- Beteiligung am Unterrichtsgespräch
- Zusammenfassungen zur Vor- und Nachbereitung des Unterrichts
- Präsentation von Arbeitsergebnissen
- Referate
- Mitarbeit in allen Arbeitsphasen

Praktische Leistungen am Computer

- Implementierung, Test und Anwendung von Informatiksystemen

Sonstige schriftliche Leistungen

- Lernerfolgsüberprüfung durch kurze schriftliche Übungen
- Schriftliche Übung dauern ca. 20 Minuten und umfassen den Stoff der letzten ca. 4–6 Stunden.
- 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, Quantität 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.

Grundsätze der Leistungsrückmeldung und Beratung

Die Grundsätze der Leistungsbewertung werden zu Beginn eines jeden Halbjahres 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.

3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen

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

4 Qualitätssicherung und Evaluation

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