



SCHULINTERNER LEHRPLAN INFORMATIK

Sekundarstufe I und II

Stand August 2022

1 Rahmenbedingungen der fachlichen Arbeit

Fachliche Bezüge zum Leitbild der Schule

In unserem Schulprogramm ist als wesentliches Ziel der Schule beschrieben, die Lernenden als Individuen mit jeweils besonderen Fähigkeiten, Stärken und Interessen in den Blick zu nehmen. Es ist ein wichtiges Anliegen, durch gezielte Unterstützung des Lernens die Potenziale jeder Schülerin und jedes Schülers in allen Bereichen optimal zu entwickeln. Das Fach Informatik leistet hierzu einen Beitrag im Bereich der digitalen Bildung.

Zur Förderung des selbstständigen und individuellen Lernens erfolgt die Arbeit im Fach Informatik nach einem gemeinsam mit den anderen Kernfächern abgestimmten Konzept.

Fachliche Bezüge zu den Rahmenbedingungen des schulischen Umfelds

Das Johann-Gottfried-Herder-Gymnasium ist ein Gymnasium mit gebundenem Ganzttag mit erweiterten Bildungsangeboten, an dem zurzeit ca.1200 Schülerinnen und Schüler von 100 Lehrpersonen unterrichtet werden.

Unterrichtliche Bedingungen

Die Fachkonferenz Informatik tritt mindestens einmal pro Schulhalbjahr zusammen, um notwendige Absprachen zu treffen. In der Regel nimmt auch ein Mitglied der Elternpflegschaft sowie der Schülervvertretung an den Sitzungen teil. Außerdem finden innerhalb der Fachgruppe zu bestimmten Aufgaben weitere Besprechungen statt.

Die Schule besitzt drei Computerräume und eine große Anzahl an iPads. Alle Räume sind mit ca. 20 Schülerrechnern, 1 Lehrerrechner und Beamern ausgestattet. In einem Computerraum befindet sich zusätzlich ein Smartboard. Alle Räume sind an das pädagogische Netzwerk angeschlossen.

Der Informatikunterricht in Klasse 6 dient dem Kennenlernen von Programmier Techniken, Algorithmen und Codierungen. Er hilft bei der Entscheidung für ein Wahlpflichtfach in Klasse 9 und 10. In der Oberstufe lassen sich in der Regel Grundkurse einrichten, die bis zum Abitur führen.

Projekte

In der Jahrgangsstufe 6 beschäftigen sich die Schüler in einer Projektphase mit dem Roboter BOB3. Die Jahrgangsstufe 9 programmiert stationäre Roboter mit Fischertechnik, die Jahrgangsstufe 10 mobile Roboter mit Fischertechnik. In der Oberstufe wird mit Lego Mindstorms EV3 gearbeitet. Alle Schüler der Informatikkurse nehmen regelmäßig mit Erfolg am *Biber-Wettbewerb* teil.

2 Entscheidungen zum Unterricht

2.1 Unterrichtsvorhaben Sekundarstufe 1

Unterrichtsvorhaben Klasse 6

Klassenarbeiten werden nicht geschrieben. Schriftliche Kontrollen kleineren Umfangs sind möglich, die Bewertung fließt in die Note zur sonstigen Mitarbeit ein.

Thema: Einführung in das Fach Informatik, Programmiertechniken, Algorithmen und Codierungen

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Kleiner ProgrammierEinstieg (a) erste Berührung mit Algorithmen (b) Kennenlernen von Programmstrukturen	Die Schülerinnen und Schüler <ul style="list-style-type: none"> • beschreiben und erläutern den Aufbau und die Arbeitsweise singulärer Rechner, • nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst, • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation • erlernen Fähigkeiten zur Programmierung von Algorithmen • vom Modell zum Programm 	Expresskurs unter code.org (Schüler – Express Course)
2. Aufbau und Wirkungsweise eines Rechners (a) Hardwarekomponenten erkennen und erklären (b) Bedeutung einzelner Hardwarekomponenten und ihr Zusammenspiel erfassen		(a) Punkte 1 bis 28 individuell bearbeiten
3. Codieren und Decodieren		Westermann „Praxis Informatik 5/6“
4. Algorithmen mit Scratch und C (a) PAP und Struktogramme		Scratch
5. Spieleprogrammierung		DrohnenSpiel aus „Praxis Informatik 5/6“ S. 56 – 76
		Projekt BOB3 (Programmiersprache C)
		Microworlds Ex Einführung

Unterrichtsvorhaben Klasse 9 – I (Wahlpflichtbereich II)

Im ersten Halbjahr 2 Klassenarbeiten, im 2. Halbjahr eine Klassenarbeit und ein Projekt. Die Länge der Klassenarbeiten beträgt 45 Minuten. Die Klassenarbeiten werden ohne Computereinsatz geschrieben.

Thema: Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Einstieg in die imperative Programmierung</p> <ul style="list-style-type: none"> (a) Algorithmen und ihre Strukturen (b) PAP und Struktogramm (c) Phasen der Programmentwicklung (d) Virtuelle Welten 	<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“, • nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst, • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation. 	<p>Hamstersimulator Dietrich Boles „Programmieren spielend gelernt mit dem Java-Hamster-Modell“</p>
<p>2. Aufbau informationsverarbeitender Systeme</p> <ul style="list-style-type: none"> • Identifikation typischer Komponenten informatischer Systeme, Herleitung der „Von-Neumann-Architektur“ • Identifikation des EVA-Prinzips (Eingabe-Verarbeitung-Ausgabe) • Binäre Berechnungen 		<p>Stationenlernen Codieren - Decodieren</p>

Unterrichtsvorhaben Klasse 9 – II

Thema: Grundlagen der Analyse, Modellierung und Implementierung anhand kleinerer Projekte

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Von der Idee zum funktionierenden Modell</p> <p>(a) Mit einem lebensnahen Beispiel werden Objekte im Sinne der Modellierung eingeführt.</p> <p>(b) Modellierung durch Flussdiagramme / Programmablaufpläne</p> <p>(c) Umsetzung von Alltagsproblemen in Modelle.</p> <p>(d) Vertiefung: Modellierung weiterer Beispiele ähnlichen Musters</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen, bauen ein passendes Robotermodell, implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (Robo Pro), stellen den Zustand eines Objekts mithilfe der stationären Roboter dar. 	<p>Fischertechnik Robo Pro Level 1</p> <p>→Einführung Interface (ppt)</p> <p>→statische Modelle: <i>Händetrockner</i></p> <p>→Arbeitsblätter vorhanden; ppt</p>
<p>2. Analyse von komplexen Problemen</p> <p>(a) Programmierung als modularisiertes Vorgehen (Entwicklung von Problemlösungen auf Grundlage vorhandener Fragestellungen)</p> <p>(b) Teilanalyse des Problems</p> <p>(c) Praktische Lösungsansätze und deren Umsetzung</p>		<p>Fischertechnik Robo Pro Level 2</p> <p>→Arbeiten mit Unterprogrammen</p> <p>Beispiel: <i>Garagentorsteuerung</i></p> <p>Beispiel: <i>Ampel1 ohne Unterprogramm, Ampel2 mit Unterprogramm</i></p> <p>Beispiel: <i>Schiebetür</i></p> <p>Arbeitsblätter vorhanden</p>
<p>3. Implementierung mit Operatoren und Variablen</p> <p>(a) Verwenden von Analogeingängen und Bedienelementen</p> <p>(b) Deklaration von Operatoren und Variablen</p>		<p>Fischertechnik Robo Pro Level 3</p> <p>→ Handbuch Kapitel 5</p> <p>Beispiel: <i>Parkhausschranke</i></p>

Unterrichtsvorhaben Klasse 9 – III

Thema: Grundlagen der objektorientierten Programmierung mit dem Hamstersimulator

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Bewegungsanimationen am Beispiel einfacher grafischer Objekte</p> <p>(a) Globale Problemlösungen (b) Spielen oder Lernen? (c) Programmstrukturen (d) Prozeduren mit und ohne Parameter</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern einfache Algorithmen und Programme, • entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar, • modifizieren einfache Algorithmen und Programme, • implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen 	<p>Hamstersimulator</p> <p>Dietrich Boles, Cornelia Boles „Objektorientierte Programmierung spielend gelernt mit dem Java – Hamster – Modell“</p>
<p>2. Einstieg in die Programmierung komplexer Projekte</p> <p>(a) Problemlösestrategien entwickeln (b) Zusammensetzen einzelner Programmmodule</p>	<ul style="list-style-type: none"> • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), • testen Programme schrittweise anhand von Beispielen • interpretieren Fehlermeldungen und korrigieren den Quellcode. 	<p>Projektideen: „Lotto Hamster“, „Hamster im Labyrinth“, „Hamster Wettrennen“ ...</p>

Unterrichtsvorhaben Klasse 9 – IV

Thema: Modellierung und Implementierung von grafischen Spielen und Simulationen (Projekt)

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Projektplanung und Durchführung (a) Nach inhaltlichen Vorgaben (b) Nach zeitlichen Vorgaben	Die Schülerinnen und Schüler <ul style="list-style-type: none"> • analysieren und erläutern Problemstellungen, • erarbeiten Schritte zur Erstellung eines Projekts • Planung und Umsetzung eines Projekts in einem vorgegebenen, • testen Programme schrittweise anhand von kleineren Modulen, • interpretieren Fehlermeldungen und korrigieren den Quellcode, • fügen Teilmodule zum fertigen Projekt zusammen 	
2. Planung eines mobilen Roboters a) Erlernen strukturierter Arbeitsweisen b) Teamfähigkeit		Fischertechnik Mobile Set <i>Hindernisserkennung</i> <i>Kantenerkennung</i> <i>Lichtsucher</i> <i>Spurensucher</i> Für Profis: Laufroboter

Unterrichtsvorhaben Klasse 10 – I. Halbjahr (Wahlpflichtbereich II)

Im ersten Halbjahr 2 Klassenarbeiten, im 2. Halbjahr eine Klassenarbeit und ein Projekt. Die Länge der Klassenarbeiten beträgt 45 Minuten. Die Klassenarbeiten werden ohne Computereinsatz geschrieben.

Thema: Programmieren für alle – Sinn und Nutzen der Programmierung von Robotern

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Was bedeutet „Programmieren“ in der modernen Informationsgesellschaft?</p> <ul style="list-style-type: none"> • Schreiben ist Macht • Schreiben mit Symbolen • Objekte, Relationen, Zustände, Handlungen 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • erkennen die Bedeutung von Programmier-techniken, setzen sich mit den wesentlichen Begriffen auseinander, • werden sich ihrer Verantwortung in einer modernen Informationsgesellschaft bewusst 	<p>Recherche: „Was versteht man unter einer „modernen Informationsgesellschaft“?“</p>
<p>2. Mobile Robots</p> <ul style="list-style-type: none"> • Robotermodelle nach Anleitung konstruieren • Steuerungssoftware erstellen • Modell und Software testen 	<ul style="list-style-type: none"> • entwickeln von der Idee über den Programmablaufplan einen Code zur Steuerung fahrbarer Modelle • implementieren komplexe Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (Lego Mindstorms), • stellen den Zustand eines Objekts mithilfe der Roboter dar. 	<p style="text-align: center;">LEGO – Mindstorms EV 3</p>

Unterrichtsvorhaben Klasse 10 – II. Halbjahr

Thema: Grundlagen der objektorientierten Programmierung, Modellierung und Implementierung anhand kleinerer Projekte

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Was bedeutet OOP</p> <p>(e) Mit einem lebensnahen Beispiel werden Objekte im Sinne der Modellierung eingeführt.</p> <p>(f) Modellierung durch Flussdiagramme / Programmablaufpläne</p> <p>(g) Umsetzung von Alltagsproblemen in Modelle.</p> <p>(h) Vertiefung: Modellierung weiterer Beispiele ähnlichen Musters</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen, implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (Greenfoot), 	<p>Arbeitsblatt „Fahrrad“</p> <p>Einführung in Greenfoot</p>
<p>2. Analyse von komplexen Problemen</p> <p>(d) Programmierung als modularisiertes Vorgehen (Entwicklung von Problemlösungen auf Grundlage vorhandener Fragestellungen)</p> <p>(e) Teilanalyse des Problems</p> <p>(f) Praktische Lösungsansätze und deren Umsetzung</p>		<p>Beispiel: <i>Little Crab, Schafe und Wölfe</i></p>
<p>3. Implementierung eines Spiels</p> <p>(c) Grundaufbau einer Java-Klasse</p> <p>(d) Deklaration und Initialisierung von Objekten</p>		<p>Projekt größeren Umfangs</p>

2.2 Unterrichtsvorhaben Sekundarstufe 2

2.2.1 Einführungsphase EF

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

Die Schülerinnen und Schüler

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

Im ersten Halbjahr 1 Klausur, im 2. Halbjahr eine Klausur und ein Projekt. Die Länge der Klausuren beträgt 90 min.

Unterrichtsvorhaben EF- I

Thema: Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten

Vorhabenbezogene Konkretisierung:

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

Es 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.

EF- I

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Information, deren Kodierung und Speicherung</p> <ul style="list-style-type: none"> • Informatik als Wissenschaft der Verarbeitung von Informationen • Darstellung von Informationen in Schrift, Bild und Ton • Speichern von Daten mit informatischen Systemen • Vereinbarung von Richtlinien zur Datenspeicherung auf den Schulrechnern (z.B. Ordnerstruktur, Dateibezeichner usw.) 	<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“, • nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst, • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation. 	<p>„Informatik macchiato“ - Cartoonkurs für Schüler, Pearson – Verlag, 2. Auflage Alltagsalgorithmen S. 54 - 55</p>
<p>2. Aufbau informatischer Systeme</p> <ul style="list-style-type: none"> • Identifikation typischer Komponenten informatischer Systeme und anschließende Beschränkung auf das Wesentliche, Herleitung der „Von-Neumann-Architektur“ <p>Identifikation des EVA-Prinzips (Eingabe-Verarbeitung-Ausgabe) als Prinzip der Verarbeitung von Daten und Grundlage der „Von-Neumann-Architektur“</p>		<p>„Informatik 5“ Klett Verlag S. 96 - 99 <i>Material:</i> Demonstrationshardware Durch Demontage eines Demonstrationsrechners entdecken Schülerinnen und Schüler die verschiedenen Hardwarekomponenten eines Informatiksystems. Als Demonstrationsrechner bietet sich ein ausrangierter Schulrechner an.</p>
<p>3. Vom Algorithmus zur Programmierung</p> <ul style="list-style-type: none"> • Programmentwicklung • Codierung • Grafische Lösungen 		<p>Programmieren mit Processing – einfache Einstiegsaufgaben aus der Programmierung zur Veranschaulichung und Umsetzung von Algorithmen https://michaelkipp.de/processing/</p>

Unterrichtsvorhaben EF- II

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

Vorhabenbezogene Konkretisierung:

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

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

Im Anschluss wird mit der Realisierung erster Projekte begonnen. Dazu muss der grundlegende Aufbau einer Java-Klasse thematisiert und zwischen Deklaration, Initialisierung und Methodenaufrufen unterschieden werden.

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

EF- II

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
Algorithmen und deren Umsetzung	Die Schülerinnen und Schüler	Programmieren mit Processing
Modellierung	<ul style="list-style-type: none"> ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen, modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen, stellen die Kommunikation zwischen Objekten grafisch dar, implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache, stellen den Zustand eines Objekts dar. 	https://michaelkipp.de/processing/
<ul style="list-style-type: none"> Mit einem lebensweltnahen Beispiel werden Objekte im Sinne der OOP Modellierung eingeführt Objekte werden mit Objektkarten visualisiert und mit sinnvollen Attributen und „Fähigkeiten“, d.h. Methoden, versehen Vertiefung: Modellierung weiterer Beispiele ähnlichen Musters 		Schöningh „Informatik“ Band 1 - 3

Unterrichtsvorhaben EF- III

Thema: Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen in Java

Vorhabenbezogene Konkretisierung:

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

Sind an einem solchen Beispiel im Schwerpunkt Schleifen und Verzweigungen eingeführt worden, sollen diese Konzepte an weiteren Beispielprojekten eingeübt werden. Dabei muss es sich nicht zwangsläufig um solche handeln, bei denen Kontrollstrukturen lediglich zur Animation verwendet werden. Auch die Erzeugung größerer Mengen grafischer Objekte und deren Verwaltung in einem Feld kann ein Anlass zur Verwendung von Kontrollstrukturen sein.

Das Unterrichtsvorhaben schließt mit einem Projekt, das komplexere grafische Elemente beinhaltet, so dass die Schülerinnen und Schüler mehr als nur die Klasse erstellen müssen, welche die Szene als Ganzes darstellt. Elemente der Szene müssen zu sinnhaften eigenen Klassen zusammengefasst werden, die dann ihre eigenen Attribute und Dienste besitzen. Auch dieses Projekt soll eine Animation, ggf. im Sinne einer Simulation, sein, bei der Attributwerte von Objekten eigener Klassen verändert werden und diese Veränderungen optisch sichtbar gemacht werden.

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

EF- III

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>Objekte und Klassen in Java</p> <ul style="list-style-type: none"> • Kontinuierliche Verschiebung eines Objekts mit Hilfe einer Schleife (While-Schleife) • Tastaturabfrage zur Realisierung einer Schleifenbedingung für eine Animationsschleife • Mehrstufige Animationen mit mehreren sequenziellen Schleifen 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern einfache Algorithmen und Programme, • entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar, • ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen, • modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen, • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu, • ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu, • modifizieren einfache Algorithmen und Programme, • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken, 	<p>Programmieren mit Processing</p> <p>https://michaelkipp.de/processing/</p> <p>Schöningh „Informatik“ Band 1 – 3</p>
<p>Erstellen und Verwalten größerer Mengen einfacher grafischer Objekte</p> <ul style="list-style-type: none"> • Erzeugung von Objekten mit Hilfe von Zählschleifen (FOR-Schleife) • Verwaltung von Objekten in eindimensionalen Feldern (Arrays) • Animation von Objekten, die in eindimensionalen Feldern (Arrays) verwaltet werden • Vertiefung: Verschiedene Feldbeispiele 	<ul style="list-style-type: none"> • implementieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen, • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu, • ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu, • modifizieren einfache Algorithmen und Programme, • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken, 	
<p>Modellierung und Animation komplexerer grafisch repräsentierbarer Objekte</p> <ul style="list-style-type: none"> • Modellierung eines Simulationsprogramms mit eigenen Klassen, die sich selbst mit Hilfe von einfachen Objekten zeigen mit Hilfe eines Implementationsdiagramms • Implementierung eigener Methoden mit und ohne Parameterübergabe • Realisierung von Zustandsvariablen 	<ul style="list-style-type: none"> • implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen, • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), • testen Programme schrittweise anhand von Beispielen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I). 	<p>Exkurs: „Guter Programmierstil“ vom → Pseudocode über → Struktogramm zum → Programmiercode</p>

Unterrichtsvorhaben EF- IV

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

Vorhabenbezogene Konkretisierung:

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

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

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

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>Vertiefung von Diagrammtypen</p> <ul style="list-style-type: none"> Objektdiagramm Klassendiagramm und Entwurfsprinzipien Entwurfsdiagramm Implementationsdiagramm Neu: Entwurfsnotation von UML-Klassendiagrammen, Multiplizitäten, Vererbung Neu: Anwendungsfalldiagramm 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> analysieren und erläutern eine objektorientierte Modellierung, stellen die Kommunikation zwischen Objekten grafisch dar, ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen, 	<p>Wiederholung von UML-Notationen</p> <p><i>Beispiel:</i> Die 3 Amigos und UML</p> <p>Material: LB Klett „Informatik 3“ S.126 – 127</p>
<p>Entwicklung eines Spiels mit der Notwendigkeit von Kollisionskontrollen zwischen zwei oder mehr grafischen Objekten</p> <ul style="list-style-type: none"> Modellierung des Spiels ohne Berücksichtigung der Kollision mit Hilfe eines Implementationsdiagramms Dokumentation der Klassen des Projekts Implementierung eines Prototypen ohne Kollision Ergänzung einer Kollisionsabfrage durch zusätzliche Assoziationsbeziehungen in Diagramm, Dokumentation und Quellcode Verallgemeinerung der neuen Verwendung von Objektreferenzen Vertiefung: Entwicklung weiterer Spiele und Simulationen mit vergleichbarer Grundmodellierung 	<ul style="list-style-type: none"> modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen, ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu, ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu, modellieren Klassen unter Verwendung von Vererbung, implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken, testen Programme schrittweise anhand von Beispielen, interpretieren Fehlermeldungen und korrigieren den Quellcode, 	<p>Programmieren mit Processing</p> <p>https://michaelkipp.de/processing/</p> <p>Schöningh „Informatik“ Band 1 – 3</p>

<p>Erarbeitung einer Simulation mit grafischen Objekten, die sich durch unterschiedliche Ergänzungen voneinander unterscheiden (Vererbung durch Spezialisierung ohne Überschreiben von Methoden)</p> <ul style="list-style-type: none"> • Analyse und Erläuterung einer Basisversion der grafischen Klasse • Realisierung von grafischen Erweiterungen zur Basisklasse mit und ohne Vererbung (Implementationsdiagramm und Quellcode) • Verallgemeinerung und Reflexion des Prinzips der Vererbung am Beispiel der Spezialisierung 	<ul style="list-style-type: none"> • modifizieren einfache Algorithmen und Programme, • stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar, • dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden. 	<p>Programmieren mit Processing</p> <p>https://michaelkipp.de/processing/</p> <p>Schöningh „Informatik“ Band 1 – 3</p>
<p>Entwicklung einer komplexeren Simulation mit grafischen Elementen, die unterschiedliche Animationen durchführen (Vererbung mit Überschreiben von Methoden)</p> <ul style="list-style-type: none"> • Analyse und Erläuterung einer einfachen grafischen Animationsklasse • Spezialisierung der Klasse zu Unterklassen mit verschiedenen Animationen durch Überschreiben der entsprechenden Animationsmethode • Reflexion des Prinzips der späten Bindung 		

Unterrichtsvorhaben EF- V

Thema: Such- und Sortieralgorithmen anhand kontextbezogener Beispiele

Vorhabenbezogene Konkretisierung:

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

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

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

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

EF- V

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 , • entwerfen einen weiteren Algorithmus zum Sortieren, • analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an. 	<p>Beispiel „Ordnung ist das halbe Leben“ aus „Informatik macchiato“ – Cartoonkurs für Schüler, Pearson-Verlag</p> <p>Beispiel „Ordnung muss sein“ aus „Abenteuer Informatik“ Kapitel 2 S.41-79</p> <p>(b) Kästen mit Kartenmaterial vorhanden</p>
<p>2. Systematisierung von Algorithmen und Effizienzbetrachtungen</p> <p>(a) Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode (auf jeden Fall: Sortieren durch Vertauschen, Sortieren durch Auswählen)</p> <p>(b) Anwendung von Sortieralgorithmen auf verschiedene Beispiele</p> <p>(c) Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche</p> <p>(d) Variante des Sortierens durch Auswählen (Nutzung eines einzigen oder zweier Felder)</p> <p>(e) Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des Speicherplatzbedarfs</p>		<p><i>Beispiele:</i> Sortieren durch Auswählen, Sortieren durch Vertauschen Quicksort (Quicksort ist als Beispiel für einen Algorithmus nach dem Prinzip <i>Teile und Herrsche</i> gut zu behandeln.)</p> <p>Kenntnisse in rekursiver Programmierung sind nicht erforderlich, da eine Implementierung nicht angestrebt wird.</p>

Unterrichtsvorhaben EF- VI

Thema: Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes

Vorhabenbezogene Konkretisierung:

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

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

EF- VI

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>Selbstständige Erarbeitung von Themen durch die Schülerinnen und Schüler</p> <p>(a) Mögliche Themen zur Erarbeitung in Kleingruppen:</p> <ul style="list-style-type: none"> • „Eine kleine Geschichte der Digitalisierung: vom Morsen zum modernen Digitalcomputer“ • „Eine kleine Geschichte der Kryptographie: von Caesar zur Enigma“ • „Von Nullen, Einsen und mehr: Stellenwertsysteme und wie man mit ihnen rechnet“ • „Kodieren von Texten und Bildern: ASCII, RGB und mehr“ • „Auswirkungen der Digitalisierung: Veränderungen der Arbeitswelt und Datenschutz“ <p>(b) Vorstellung und Diskussion durch Schülerinnen und Schüler</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen, • erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung, • stellen ganze Zahlen und Zeichen in Binärcodes dar, • interpretieren Binärcodes als Zahlen und Zeichen, • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation. 	<p><i>Beispiel:</i> Ausstellung zu informatischen Themen</p> <p>Die Schülerinnen und Schüler bereiten eine Ausstellung zu informatischen Themen vor. Dazu werden Stellwände und Plakate vorbereitet, die ggf. auch außerhalb des Informatikunterrichts in der Schule ausgestellt werden können.</p> <p><i>Materialien:</i> Schülerinnen und Schüler recherchieren selbstständig im Internet, in der Schulbibliothek, in öffentlichen Bibliotheken, usw.</p> <p>Film „Geschichte der Computer“ Film „Konrad Zuse“</p>
<p>Erarbeitung grundlegender Begriffe des Datenschutzes</p> <p>(a) Problematisierung und Anknüpfung an die Lebenswelt der Schülerinnen und Schüler</p> <p>(b) Diskussion und Bewertung von Fallbeispielen aus dem Themenbereich „Datenschutz“</p>		<p><i>Beispiel:</i> Fallbeispiele aus dem aktuellen Tagesgeschehen</p> <p>Die Schülerinnen und Schüler bearbeiten Fallbeispiele aus ihrer eigenen Erfahrungswelt oder der aktuellen Medienberichterstattung.</p> <p><i>Materialien:</i> Materialblatt zum Bundesdatenschutzgesetz</p>

2.2.2 Qualifikationsphase Q1

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

Die Schülerinnen und Schüler

- verwenden 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).

Im ersten und im zweiten Halbjahr werden 2 Klausuren geschrieben. Im 2.Halbjahr ist eine Facharbeit statt einer Klausur möglich. Ein Projekt statt Klausur wird auch hier für das 2.Halbjahr empfohlen. Die Länge der Klausuren beträgt 90 min.

Unterrichtsvorhaben Q1- I:

Thema: Wiederholung der objektorientierten Modellierung und Programmierung

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 Schüler 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.

Q1- I

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Wiederholung und Erweiterung der objektorientierten Modellierung und Programmierung durch Analyse und Erweiterung eines kontextbezogenen Beispiels</p> <p>(a) Analyse der Problemstellung (b) Analyse der Modellierung (Implementationsdiagramm) (c) Erweiterung der Modellierung im Implementationsdiagramm (Vererbung, abstrakte Klasse) (d) Kommunikation zwischen mindestens zwei Objekten (grafische Darstellung) (e) Dokumentation von Klassen (f) Implementierung der Anwendung oder von Teilen der Anwendung</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern objektorientierte Modellierungen, • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen, • modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten, • ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu, • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren, • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken, • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen, wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an, • interpretieren Fehlermeldungen und korrigieren den Quellcode, • stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar, • dokumentieren Klassen, • stellen die Kommunikation zwischen Objekten grafisch dar. 	<p>Beispiel „Heldenspiel“</p> <p>Schöningh „Informatik“ Band 1 – 3</p>

Unterrichtsvorhaben Q1- II:

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

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 entsprechende Oberfläche sowie die Klasse `Queue` wird dabei von der Lehrkraft vorgegeben. Anschließend wird die Anwendung modifiziert, um den Umgang mit der Datenstruktur zu üben. Anhand einer Anwendung, in der Daten nach dem Last-In-First-Out-Prinzip verwaltet werden, werden Unterschiede zwischen den Datenstrukturen Schlange und Stapel erarbeitet. Um einfacher an Objekte zu gelangen, die zwischen anderen gespeichert sind, wird die Klasse `List` eingeführt und in einem Anwendungskontext verwendet. In mindestens einem weiteren Anwendungskontext wird die Verwaltung von Daten in Schlangen, Stapeln oder Listen vertieft. Modellierungen werden dabei in Entwurfs- und Implementationsdiagrammen dargestellt.

Q1- II

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>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen, • analysieren und erläutern Algorithmen und Programme, • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen, • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nicht-lineare Datensammlungen zu, • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen, • modifizieren Algorithmen und Programme, • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen, • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen, • interpretieren Fehlermeldungen und korrigieren den Quellcode, • testen Programme systematisch anhand von Beispielen, • stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau 	<p>Klett Informatik 4 S.10 – 15</p> <p><i>Beispiel: Warteschlangen</i></p> <p>konkret: Taxiwarteschlange in Processing mit Feldern</p> <p>Anwendung: Schülerschlange am Kiosk mit Feldern</p> <p>Schöningh „Informatik“ Band 1 – 3</p>

	<ul style="list-style-type: none"> • vergleichen die Datenstrukturen Warteschlange und Stapel 	
<p>2. Die Datenstruktur Liste im Anwendungskontext unter Nutzung der Klasse List</p> <p>(a) Erarbeitung der Vorteile der Klasse List im Gegensatz zu den bereits bekannten linearen Strukturen</p> <p>(b) Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse List.</p> <p>(c) Rekursive Funktionen</p> <p>(d) Rekursive Methoden einer Liste</p>	<ul style="list-style-type: none"> • sind mit dem LIFO- und FIFO-Prinzip vertraut 	<p>Klett Informatik 4 S.16 – 31</p> <p><i>Beispiel: Warteschlangen</i></p> <p>konkret: Taxiwarteschlange in Processing mit Listen</p> <p>Anwendung: Schülerschlange am Kiosk mit Listen</p> <p><i>Beispiel:</i></p> <ul style="list-style-type: none"> ➔ Kaninchenproblem (Fibonacci) ➔ Reiskornparabel (Film)
<p>3. 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>Klett Informatik 4 S.32 – 36</p> <p><i>Beispiel:</i></p> <p>UNO – ein beliebtes Kartenspiel</p> <p>Der Eisbär</p> <p>Schwerpunkt:</p> <ul style="list-style-type: none"> ➔ Türme von Hanoi ➔ (www.blinde-kuh.de/spiele/hanoi)
<p>(a) Vertiefung - Anwendungen von Listen, Stapeln oder Schlangen</p>		<p><i>Beispiel:</i></p> <p>Der Biber im Restaurant (aus Schöningh „Informatik 2“ S.73)</p>

Unterrichtsvorhaben Q1- III:

Thema: „Ich habe ein Problem ..., denn Ordnung muss sein!“

Vorhabenbezogene Konkretisierung:

Zuerst wird die Problematik von Bedienungsanleitungen, Sprache als Beschreibung und Überführung in konkrete Algorithmen behandelt und vertieft. Hierbei wird der Einsatz von Struktogrammen und Programmablaufplänen wiederholt, Kontrollstrukturen vertieft.

In einem Anwendungskontext werden 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.

In diesem Lernabschnitt steht das eigenverantwortliche Arbeiten (kurz EVA) im Vordergrund. Durch Ausprobieren sollen Sortierverfahren erkannt und umgesetzt werden. Die neuen Erkenntnisse werden algorithmisch formuliert und vom Pseudocode über das Struktogramm in einen Java Code überführt.

Q1- III

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Algorithmen – Methoden und Struktur</p> <p>(a) Bedienungsanleitungen (b) Merkmale von Algorithmen (c) Pseudocode – Struktogramm/PAP – Implementierung (d) Vertiefung der Kontrollstrukturen (e) Wie löst man komplexe Probleme?</p> <p>2. Suchen von Daten in Listen und Arrays</p> <p>(f) Lineare Suche in Listen und in Arrays (g) Binäre Suche in Arrays als Beispiel für rekursives Problemlösen (h) Untersuchung der beiden Suchverfahren hinsichtlich ihrer Effizienz (Laufzeitverhalten, Speicherbedarf)</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> analysieren und erläutern Algorithmen und Programme, beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen, beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen, entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“, modifizieren Algorithmen und Programme, implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen, implementieren und erläutern iterative und rekursive Such- und Sortierverfahren, nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen, interpretieren Fehlermeldungen und korrigieren den Quellcode, testen Programme systematisch anhand von Beispielen, 	<p>Klett Informatik 3 S.17 – 24</p> <p>Kontrollstrukturen aus: <i>Schöningh „Informatik 1“ S.59 – 69 neue Ausgabe</i> → „Der Rover hat ein Problem“</p> <p><i>Schöningh „Informatik 2“ S.94 – 95</i> → Projekteinstieg 1 „Suchen“</p>
<p>3. 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 (b) Implementierung eines einfachen Sortierverfahrens für ein Feld (c) Entwicklung eines rekursiven Sortierverfahrens für ein Feld</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> analysieren und erläutern Algorithmen und Programme, beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen, beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen, entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“, modifizieren Algorithmen und Programme, implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen, implementieren und erläutern iterative und rekursive Such- und Sortierverfahren, nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen, interpretieren Fehlermeldungen und korrigieren den Quellcode, testen Programme systematisch anhand von Beispielen, 	<p><i>Schöningh „Informatik 2“ S.96 – 97</i> → Projekteinstieg 2 „Sortieren“</p> <p>Projekt: „Suchen“ in Processing</p>

Unterrichtsvorhaben Q1- IV:

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

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 \rightarrow 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.

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, • analysieren und erläutern Algorithmen und Programme, • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen, • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen, • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu, • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren, • verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie, • entwickeln iterative und rekursive Algorithmen unter Nutzung der Kon- 	<p><i>Beispiel:</i> Ahnenbaum</p> <p>Die binäre Baumstruktur ergibt sich daraus, dass jede Person genau einen Vater und eine Mutter hat.</p> <p>Schöningh „Informatik“ Band 1 – 3</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> <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><i>Beispiel:</i> Informatikerbaum als binärer Baum</p> <p>In einem <i>binären Baum</i> werden die Namen und die Geburtsdaten von Informatikern lexikographisch geordnet abgespeichert. Alle Namen, die nach dieser Ordnung vor dem Namen im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem Namen im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.)</p> <p>Folgende Funktionalitäten werden benötigt:</p> <ul style="list-style-type: none"> • Einfügen der Informatiker-Daten in den Baum • Suchen nach einem Informatiker über den Schlüssel Name • Ausgabe des kompletten Datenbestands in nach Namen sortierter Reihenfolge

<p>(e) Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf</p>	<p>struktionsstrategien „Modularisierung“ und „Teilen und Herrschen“,</p>	
<p>3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse BinarySearchTree</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramm, grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften</p> <p>(c) Erarbeitung der Klasse BinarySearchTree und Einführung des Interface Item zur Realisierung einer geeigneten Ordnungsrelation</p> <p>(d) Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums</p>	<ul style="list-style-type: none"> • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen, • modifizieren Algorithmen und Programme, • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen, • interpretieren Fehlermeldungen und korrigieren den Quellcode, • testen Programme systematisch anhand von Beispielen, • stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau, • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar. 	<p><i>Beispiel:</i> Informatikerbaum als <i>Suchbaum</i></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>4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen</p>		<p>Schöningh „Informatik“ Band 1 -3</p> <ul style="list-style-type: none"> ➔ Projekteinstieg 1: Spielbäume ➔ Projekteinstieg 2: Binäre Suchbäume ➔ Projekteinstieg 3: Navigationssysteme

Unterrichtsvorhaben Q1- V:

Thema: Endliche Automaten und formale Sprachen

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.

Q1- V

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p>1. Endliche Automaten</p> <p>(a) Vom Automaten in den Schülerinnen und Schülern bekannten Kontexten zur formalen Beschreibung eines endlichen Automaten</p> <p>(b) Untersuchung, Darstellung und Entwicklung endlicher Automaten</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens auf bestimmte Eingaben, • analysieren und erläutern Grammatiken regulärer Sprachen, • zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf, • ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird, 	<p><i>Beispiele:</i> Cola-Automat, Geldspielautomat, Roboter, Zustandsänderung eines Objekts „Auto“, Akzeptor für bestimmte Zahlen, Akzeptor für Teilwörter in längeren Zeichenketten, Akzeptor für Terme</p>
<p>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>	<ul style="list-style-type: none"> • entwickeln und modifizieren zu einer Problemstellung endliche Automaten, • entwickeln und modifizieren zu einer Problemstellung endliche Automaten, • entwickeln zur akzeptierten Sprache eines Automaten die zugehörige Grammatik, • entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten, • modifizieren Grammatiken regulärer Sprachen, • entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt, • stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform, • ermitteln die Sprache, die ein endlicher Automat akzeptiert. 	<p><i>Beispiele:</i> reguläre Grammatik für Wörter mit ungerader Parität, Grammatik für Wörter, die bestimmte Zahlen repräsentieren, Satzgliederungsgrammatik</p>
<p>3. Grenzen endlicher Automaten</p>		<p>J – Flap</p> <p>flaci - online</p>

2.2.3 Qualifikationsphase Q2

Es sind zwei Klausuren vorgegeben. Die Länge der Klausur beträgt 135 min. Ein Vorabitur findet gemäß Abiturvorgaben statt.

Unterrichtsvorhaben Q2 – I:

Thema: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

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.

Q2- I

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, • analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage, • analysieren und erläutern eine Datenbankmodellierung, • erläutern die Eigenschaften normalisierter Datenbankschemata, • bestimmen Primär- und Sekundärschlüssel, • ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten, • modifizieren eine Datenbankmodellierung, • modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema, • bestimmen Primär- und Sekundärschlüssel, • überführen Datenbankschemata in vorgegebene Normalformen, • verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren, • ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen, • 	<p>EVA – Eigenverantwortliches Arbeiten</p> <p>Vieweg/Teubner „Informatik – Manga Datenbanken“</p>

2. Modellierung von relationalen Datenbanken

(a) Entity-Relationship-Diagramm

- 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

(b) Entwicklung einer Datenbank aus einem Datenbankentwurf

- Modellierung eines relationalen Datenbankschematas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln

(c) Redundanz, Konsistenz und Normalformen

- 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)

- stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar,
- überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften.

SQL – ZOO online
sqlzoo.net/wiki/SQL_Tutorial

Datenbankabfragen mit Java

Projekt „Abi – Motto“

Unterrichtsvorhaben Q2- II:

Thema: Kommunikation in Netzwerken

Die Schüler vollziehen einen vereinfachten Aufbau eines Netzwerks in einem Rollenspiel nach. Dabei wächst das Netz von lokalen Netzen (LAN), bei dem die Kommunikationspartner einander direkt „sehen“, zu einem weltweiten Netz (WAN). Hier ist es nötig, dass die Weiterleitung der Nachrichten über mehrere Zwischenstationen läuft. Aus Gründen der didaktischen Reduktion werden Funktionen von Routern und Switches hier vermischt. Diese Unterscheidung ist aber nicht Ziel des Rollspiels. Sinn ist vielmehr, den Schülern die prinzipielle Wirkungsweise des Internets (Routing, Redundanz) zu zeigen.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Rollenspiel Netzwerk 2. Grundlagen der Datenübertragung (a) Topologien von Netzwerken (b) Protokolle 3. Netzwerke selbst erstellen	Die Schülerinnen und Schüler <ul style="list-style-type: none"> wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an, untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen 	Rollenspiel Netzwerke aus https://lehrerfortbildung-bw.de Java – Applikation <i>New Town</i> und <i>Gold City</i> FILIUS – Freie Interaktive Lernumgebung für Internetworking

Unterrichtsvorhaben Q2- III:

Thema: Sicherheit und Datenschutz in Netzstrukturen

Vorhabenbezogene Konkretisierung:

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

Q2- III

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Daten in Netzwerken und Sicherheitsaspekte in Netzen sowie beim Zugriff auf Datenbanken</p> <p>(a) Beschreibung eines Datenbankzugriffs im Netz anhand eines Anwendungskontextes und einer Client-Server-Struktur zur Klärung der Funktionsweise eines Datenbankzugriffs</p> <p>(b) 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>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • untersuchen und bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen, die Sicherheit von Informatiksystemen sowie die Einhaltung der Datenschutzbestimmungen und des Urheberrechts, • untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen, • nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte. 	<p>Verschiedene Fernsehdokumentationen</p> <ul style="list-style-type: none"> ➔ „Gestohlene Identität“ (MDR) ➔ „Krieg der Drohnen“ (ARTE) ➔ „NetWars – Krieg im Netz“ (ARTE)“ <p>Miniworkshop Kryptographie</p> <p>Cryptotool unter https://www.cryptool.org/de/</p>
<p>2. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht</p>		

Unterrichtsvorhaben Q2- IV:

Thema: Eigenverantwortliches Arbeiten

Vorhabenbezogene Konkretisierung:

Die Schüler transferieren erworbenes Wissen selbständig auf neue unbekannte Problemstellungen. Sie wählen selbständig eines der vorgegebenen Themen aus.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>3 D Programmierung mit Processing</p> <p>Spieleprogrammierung mit Processing</p>	<p>Wiederholung und Vertiefung ausgewählter Kompetenzen und Inhalte der Qualifikationsphase zur Vorbereitung auf das Abitur</p>	<p>Processing</p>

3 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

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

Überfachliche Grundsätze:

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

Fachliche Grundsätze:

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

4 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

Auf der Grundlage von §13 - §16 der APO-GOST sowie Kapitel 3 des Kernlehrplans Informatik für die gymnasiale Oberstufe wird die Fachkonferenz des Johann-Gottfried-Herder-Gymnasiums im Einklang mit dem entsprechenden schulbezogenen Konzept die nachfolgenden Grundsätze zur Leistungsbewertung und Leistungsrückmeldung beschließen. 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. Gleiches gilt für den Wahlpflichtbereich I der Jahrgangsstufen 8 und 9.

4.1 Beurteilungsbereich schriftliche Leistungen/Klassenarbeiten/Klausuren

Verbindliche Absprachen:

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

Dauer und Anzahl der Klassenarbeiten/Klausuren (vgl. APO SI VV zu §6)
Innerhalb des vorgegebenen Rahmens hat die Fachkonferenz folgende Festlegungen getroffen.

Klasse			
	Anzahl	Dauer (in Minuten)	Hinweise
6	0	keine	keine
9	3 + 1 Projekt	45	ein Projekt verpflichtend ohne Computer (ausgenommen das Projekt)
10	3 + 1 Projekt	45	ein Projekt verpflichtend ohne Computer (ausgenommen das Projekt)
EF	2 + 1 Projekt	90	ein Projekt verpflichtend ohne Computer (ausgenommen das Projekt)
Q1	3 + 1 Projekt	90	ein Projekt fakultativ ohne Computer (ausgenommen das Projekt)
Q2	2	135	ohne Computer

Die Bewertung der schriftlichen Leistungen in Klassenarbeiten und Klausuren erfolgt über ein Raster mit Punkten, die im Erwartungshorizont den einzelnen Kriterien zugeordnet sind. Spätestens ab der Qualifikationsphase orientiert sich die Zuordnung der Punktsumme 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 Punkte 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 Punkte erteilt werden.

4.2 Beurteilungsbereich „Sonstige Leistungen“:

- Entsprechend der Vorgaben zur Bewertung „Sonstiger Leistungen“ in Sek 1 und Sek 2 auf der Homepage.

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

Verbindliche Absprachen der Fachgruppe:

- Alle Schülerinnen und Schüler führen in allen Jahrgängen in Kleingruppen ein Kurzprojekt durch und fertigen dazu eine Dokumentation an. Dies wird in die Note für die Sonstige Mitarbeit einbezogen.
- In der Qualifikationsphase erstellen, dokumentieren und präsentieren die Schülerinnen und Schüler in Kleingruppen ein anwendungsbezogenes Softwareprodukt. Dies wird in die Note für die Sonstige Mitarbeit einbezogen.

Leistungsaspekte

Mündliche Leistungen

- Beteiligung am Unterrichtsgespräch
- Zusammenfassungen 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

- Portfolio / Dokumentation oder Lerntagebuch zu einem durchgeführten Unterrichtsvorhaben
- Lernerfolgsüberprüfung durch kurze schriftliche Übungen
- Schriftliche Übungen dauern ca. 20 Minuten und umfassen den Stoff der letzten ca. 4–6 Stunden
- Bearbeitung von schriftlichen Aufgaben im Unterricht

4.3 Bewertungskriterien

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

Die Bewertungskriterien stützen sich auf:

- Qualität der Beiträge
- Quantität der Beiträge
- Kontinuität der Beiträge

Besonderes Augenmerk ist dabei auf

- Sachliche Richtigkeit
- Angemessene Verwendung der Fachsprache
- Darstellungskompetenz
- Komplexität/Grad der Abstraktion
- Selbstständigkeit im Arbeitsprozess
- Einhaltung gesetzter Fristen
- Präzision
- Differenziertheit der Reflexion zu legen.

Bei Gruppenarbeiten auch auf

- Einbringen in die Arbeit der Gruppe
- Durchführung fachlicher Arbeitsanteile und
- die Qualität des entwickelten Produkts

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

4.4 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.

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.

5 Lehr- und Lernmittel

5.1 Bausätze

- Fischertechnik Starter - Set
- Fischertechnik Mobile - Set

- Lego Mindstorms Education EV 3 Basis - Set (45544)
- Lego Mindstorms Education EV 3 Ergänzungs - Set (45560)

5.2 Software

- RoboPro (Fischertechnik)
- Lego Mindstorms Education EV 3
- Scratch
- XLogo4Schools
- MicroWorlds Ex
- Hamstersimulator
- BluJ
- Greenfoot
- Java-Editor
- Processing
- BOB3
- Blender 3 D
- Filius
- JFlap, flaci
- Struktogrammeditor
- XAMPP Control Panel

5.3 Lehrwerke

Scratch - Spielend programmieren lernen	KnowWare Basics! Verlag
Einführung in die Programmierung mit LOGO	Springer Vieweg Verlag
START IT Band	Klett Verlag
START IT Band 2	Klett Verlag
INFORMATIK Band 2 Tabellenkalkulation / Datenbanken	Klett Verlag
INFORMATIK Band 3 Algorithmen / OOP / Zustandsmodellierung	Klett Verlag
INFORMATIK Band 4 Rekursive Datenstrukturen / Softwaretechnik	Klett Verlag
INFORMATIK Band 5 Formale Sprachen / Kommunikation und Synchronisation von Prozessen Funktionsweise eines Rechners / Grenzen der Berechenbarkeit	Klett Verlag
Programmieren spielend gelernt mit dem Java-Hamster-Modell	Springer Vieweg Verlag
Objektorientierte Programmierung spielend gelernt mit dem Java-Hamster-Modell	Springer Vieweg Verlag
Abenteuer Informatik IT zum Anfassen – von Routenplaner bis Online-Banking	Springer Spektrum Verlag
Informatik – Manga Datenbanken	Vieweg Teubner Verlag

6 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen

- geplante Teilnahme an der First Lego League
- Teilnahme Biber Wettbewerb
- Organisation von Programmierworkshops

7 Qualitätssicherung und Evaluation

Maßnahmen der fachlichen Qualitätssicherung:

Das Fachkollegium überprüft kontinuierlich, inwieweit die im schulinternen Lehrplan vereinbarten Maßnahmen zum Erreichen der im Kernlehrplan vorgegebenen Ziele geeignet sind. Dazu dienen beispielsweise auch der regelmäßige Austausch sowie die gemeinsame Konzeption von Unterrichtsmaterialien, welche hierdurch mehrfach erprobt und bezüglich ihrer Wirksamkeit beurteilt werden.

Alle Fachkollegen (ggf. auch die gesamte Fachschaft) nehmen regelmäßig an Fortbildungen teil, um fachliches Wissen zu aktualisieren und pädagogische sowie didaktische Handlungsalternativen zu entwickeln. Zudem werden die Erkenntnisse und Materialien aus fachdidaktischen Fortbildungen und Implementationen zeitnah in der Fachgruppe vorgestellt und für alle verfügbar gemacht.

Feedback von Schülerinnen und Schülern wird als wichtige Informationsquelle zur Qualitätsentwicklung des Unterrichts angesehen. Sie sollen deshalb Gelegenheit bekommen, die Qualität des Unterrichts zu evaluieren. Dafür kann das Online-Angebot SEFU (Schüler als Experten für Unterricht) genutzt werden (www.sefu-online.de.)

Überarbeitungs- und Planungsprozess:

Eine Evaluation erfolgt jährlich. In den Dienstbesprechungen der Fachgruppe zu Schuljahresbeginn werden die Erfahrungen des vorangehenden Schuljahres ausgewertet und diskutiert sowie eventuell notwendige Konsequenzen formuliert. Nach der jährlichen Evaluation finden sich die Jahrgangsstufenteams zusammen und arbeiten die Änderungsvorschläge für den schulinternen Lehrplan ein. Insbesondere verständigen sie sich über alternative Materialien und Kontexte der einzelnen Unterrichtsvorhaben.

Zielsetzung: Der schulinterne Lehrplan ist als „dynamisches Dokument“ zu sehen. Dementsprechend sind die dort getroffenen Absprachen stetig zu überprüfen, um ggf. Modifikationen vornehmen zu können. Die Fachschaft trägt durch diesen Prozess zur Qualitätsentwicklung und damit zur Qualitätssicherung des Faches bei.

Prozess: Die Überprüfung erfolgt jährlich. Zu Schuljahresbeginn werden die Erfahrungen des vergangenen Schuljahres in der Fachkonferenz ausgetauscht, bewertet und eventuell notwendige Konsequenzen formuliert.