

**Schulinterner Lehrplan
zum Kernlehrplan für die gymnasiale Oberstufe
der Erzbischöflichen Liebfrauenschule Köln**

Informatik

(Stand: 03.03.2017)

2.1.1 Übersichtsraster Unterrichtsvorhaben

I) Einführungsphase

Einführungsphase	
<p><u>Unterrichtsvorhaben EF-I</u></p> <p>Thema: <i>Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none">• Argumentieren• Darstellen und Interpretieren• Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none">• Informatiksysteme• Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none">• Einzelrechner• Dateisystem• Internet• Einsatz von Informatiksystemen	<p><u>Unterrichtsvorhaben EF-II</u></p> <p>Thema: <i>Grundlagen der objektorientierten Analyse, Modellierung und Implementierung anhand von einfachen Beispielkontexten</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none">• Modellieren• Implementieren• Darstellen und Interpretieren• Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none">• Daten und ihre Strukturierung• Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none">• Objekte und Klassen• Syntax und Semantik einer Programmiersprache

Einführungsphase

Unterrichtsvorhaben EF-III

Thema:

Objektorientierte Analyse, Modellierung und Implementation anhand von komplexen Beispielkontexten

Zentrale Kompetenzen:

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

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Unterrichtsvorhaben EF-IV

Thema:

Such- und Sortieralgorithmen anhand kontextbezogener Beispiele

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Daten und ihre Strukturierung
- Algorithmen

Inhaltliche Schwerpunkte:

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

Einführungsphase

Unterrichtsvorhaben EF-V

Thema:

Die digitale Welt und die Grundlagen des Datenschutzes

Zentrale Kompetenzen:

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatik, Mensch und Gesellschaft
- Informatiksysteme

Inhaltliche Schwerpunkte:

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

II) Qualifikationsphase (Q1 und Q2) - GRUNKURS

Qualifikationsphase 1

Unterrichtsvorhaben Q1-I

Thema:

Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung

Zentrale Kompetenzen:

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

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Analyse, Entwurf und Implementierung von Algorithmen
- Syntax und Semantik einer Programmiersprache
- Nutzung von Informatiksystemen

Unterrichtsvorhaben Q1-II

Thema:

Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen

Zentrale Kompetenzen:

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

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Qualifikationsphase 1

Unterrichtsvorhaben Q1-III

Thema:

Suchen und Sortieren auf linearen Datenstrukturen

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Algorithmen
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

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

Unterrichtsvorhaben Q1-IV

Thema:

Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

Zentrale Kompetenzen:

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

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Qualifikationsphase 1

Unterrichtsvorhaben Q1-V

Thema:

Sicherheit und Datenschutz in Netzstrukturen

Zentrale Kompetenzen:

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

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

Qualifikationsphase 2

Unterrichtsvorhaben Q2-I

Thema:

Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen

Zentrale Kompetenzen:

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

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Unterrichtsvorhaben Q2-II

Thema:

Endliche Automaten und formale Sprachen

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Endliche Automaten und formale Sprachen

Inhaltliche Schwerpunkte:

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

Qualifikationsphase 2

Unterrichtsvorhaben Q2-III

Thema:

Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

Zentrale Kompetenzen:

- Argumentieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Einzelrechner und Rechnernetzwerke
- Grenzen der Automatisierung

Unterrichtsvorhaben Q2-IV:

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

Unterrichtsvorhaben EF-I

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Information, deren Kodierung und Speicherung</p> <ol style="list-style-type: none">1. Informatik als Wissenschaft der Verarbeitung von Informationen2. Darstellung von Informationen in Schrift, Bild und Ton3. Speichern von Daten mit Hilfe informatischer Systeme, z.B. Schulrechner4. 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">• nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D)• nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K)• beschreiben und erläutern den Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A)
<p>2. Informations- und Datenübermittlung in Netzen</p> <ol style="list-style-type: none">1. „Sender-Empfänger-Modell“ und seine Bedeutung für die Eindeutigkeit von Kommunikation2. Informatische Kommunikation in Rechnernetzen am Beispiel des Schulnetzwerks (z.B. Benutzeranmeldung, Netzwerkordner, Zugriffsrechte, Client-Server)3. Grundlagen der technischen Umsetzung von Rechnerkommunikation am Beispiel des Internets (z.B. Netzwerkadresse, Paketvermittlung, Protokoll)4. Richtlinien zum verantwortungsvollen Umgang mit dem Internet	
<p>3. Aufbau informatischer Systeme</p>	

- | | |
|--|--|
| <ol style="list-style-type: none">1. Identifikation typischer Komponenten informatischer Systeme und anschließende Beschränkung auf das Wesentliche, Herleitung der „Von-Neumann-Architektur“2. Identifikation des EVA-Prinzips (Eingabe-Verarbeitung-Ausgabe) als Prinzip der Verarbeitung von Daten und Grundlage der Von-Neumann-Architektur | |
|--|--|

Unterrichtsvorhaben EF-II

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Identifikation von Objekten</p> <ol style="list-style-type: none"> 1. Am Beispiel eines lebensnahen Beispiels werden Objekte im Sinne der objektorientierten Modellierung eingeführt 2. Manche Objekte sind prinzipiell typgleich und werden so zu einer Objektsorte bzw. Klasse zusammengefasst 3. Objekte werden mit Klassendiagrammen visualisiert und mit sinnvollen Attributen und „Fähigkeiten“, d.h. Methoden, versehen 4. Vertiefung: Modellierung weiterer Beispiele ähnlichen Musters 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften und ihre Operationen (M) • modellieren Klassen mit ihren Attributen und Methoden (M), • stellen Attribute und Methoden in einem Klassendiagramm dar (D), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen zu (M), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • stellen den Zustand eines Objektes dar (D), • analysieren und erläutern einfache Algorithmen und Programme (A), • entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch (PAP) dar (M), • implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I), • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I).
<p>2. Implementation einfacher Klassen mit BlueJ</p> <ol style="list-style-type: none"> 1. Grundaufbau einer Java Klasse 2. Deklaration und Initialisierung von Objekten 3. Konstruktoren 4. einfache Methodenaufrufe 5. primitive Datentypen als Attribute, Parameter und Variablen 6. bedingte Anweisungen (IF-Anweisungen) 	
<p>3. Einfache Greenfoot-Szenarien</p> <ol style="list-style-type: none"> 1. Analyse von vorgegebenen Klassen 2. Grundlagen der Vererbung 3. kontinuierliches Bewegen eines Objekts mit Hilfe einer Schleife (WHILE-Schleife) 4. definiertes Bewegen eines Objektes mit Hilfe von Zählschleifen (FOR-Schleife) 5. Abfragen von Zuständen 	

Unterrichtsvorhaben EF-III

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Entwicklung von Spielen oder Simulationen mit Interaktion zwischen unterschiedlichen Objekten in Greenfoot</p> <ol style="list-style-type: none"> 1. schrittweise Implementation der Projekte 2. Dokumentation der Klassen des Projekts 3. Kollisionsabfragen 4. Verwendung von Objektreferenzen 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern eine objektorientierte Modellierung (A), • stellen die Kommunikation zwischen Objekten in Diagrammen grafisch dar (M), • ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M), • ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M), • modellieren Klassen unter Verwendung von Vererbung (M), • implementieren Klassen auch unter Nutzung dokumentierter Klassenbibliotheken (I), • testen Programme schrittweise anhand von Beispielen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • modifizieren einfache Algorithmen und Programme (I), • stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D).
<p>2. Vertiefung der Vererbung</p> <ol style="list-style-type: none"> 1. Analyse und Erläuterung einer Basis-Version einer Klasse 2. Realisierung von Erweiterungen zur Basisklasse mit Vererbung 3. Verallgemeinerung und Reflexion des Prinzips der Vererbung am Beispiel der Spezialisierung 4. Überschreiben von Methoden 5. abstrakte Oberklassen 	
<p>3. Entwicklung von BlueJ-Projekten mit linearen Datensammlungen</p> <ol style="list-style-type: none"> 1. Deklaration und Initialisierung von eindimensionalen Feldern 2. Zugriff auf die Elemente eines eindimensionalen Feldes 3. Anzahl der Elemente eines eindimensionalen Feldes 4. Darstellung eindimensionaler Felder in Klassendiagrammen 5. eindimensionale Felder als Attribute, Variablen, Parameter und Rückgaben von Methoden 6. die Klasse <code>Math</code> 7. die Klasse <code>String</code> 	

Unterrichtsvorhaben EF-IV

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
1. Explorative Erarbeitung eines Sortierverfahrens <ol style="list-style-type: none">Sortierprobleme im Kontext informatischer Systeme und im Alltag (z.B. Dateisortierung, Tabellenkalkulation, Telefonbuch, Bundesligatabelle usw.)Vergleich zweier Elemente als Grundlage eines Sortieralgorithmusselbstständige Erarbeitung eines Sortieralgorithmus	Die Schülerinnen und Schüler <ul style="list-style-type: none">beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeit und Speicherplatzbedarf (A),entwerfen einen weiteren Algorithmus zum Sortieren (M),analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D).
2. Systematisierung von Algorithmen und Effizienzbetrachtungen <ol style="list-style-type: none">Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode (auf jeden Fall: Sortieren durch Vertauschen, Sortieren durch Auswählen)Anwendung von Sortieralgorithmen auf verschiedene BeispieleBewertung von Algorithmen anhand der Anzahl der notwendigen VergleicheVariante des Sortierens durch Auswählen (Nutzung eines einzigen oder zweier Felder, bzw. lediglich eines einzigen zusätzlichen Ablageplatzes oder mehrerer neuer Ablageplätze)Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des SpeicherplatzbedarfsAnalyse des weiteren Sortieralgorithmus (sofern nicht in Sequenz 1 und 2 bereits geschehen)	
3. Binäre Suche auf sortierten Daten	

- | | |
|--|--|
| <ol style="list-style-type: none">1. Suchaufgaben im Alltag und im Kontext informatischer Systeme2. evtl. Simulationsspiel zum effizienten Suchen mit binärer Suche3. Effizienzbetrachtungen zur binären Suche | |
|--|--|

Unterrichtsvorhaben EF-V

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Informatiker verändern die Welt – Auswirkungen der Informationstechnologie auf Mensch und Gesellschaft</p> <p>Die digitale Welt 001 – Von analog zu digital 010 – Binäre Welt 011 – Der von-Neumann-Rechner 100 – Von der Schrift zum Smartphone 101 – Leben in der digitalen Welt 111 – Automatisierung</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">• stellen ganze Zahlen und Zeichen in Binärcodes dar (D),• interpretieren Binärcodes als Zahlen und Zeichen (D),• bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A),• erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A),• nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K).
<p>2. Datenschutz – Der gläserne Mensch</p>	

II) Qualifikationsphase

Unterrichtsvorhaben Q1-I:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Wiederholung und Erweiterung der objektorientierten Modellierung und Programmierung durch Analyse und Erweiterung eines kontextbezogenen Beispiels</p> <ol style="list-style-type: none">1. Analyse der Problemstellung2. Analyse der Modellierung (Implementationsdiagramm)3. Erweiterung der Modellierung im Implementationsdiagramm (Vererbung, abstrakte Klasse)4. Kommunikation zwischen mindestens zwei Objekten (grafische Darstellung)5. Dokumentation von Klassen6. Implementierung der Anwendung oder von Teilen der Anwendung	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">• analysieren und erläutern objektorientierte Modellierungen (A),• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),• modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M),• ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M),• modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),• implementieren Klassen auch unter Nutzung dokumentierter Klassenbibliotheken (I),• nutzen die Syntax und Semantik 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 mit Diagrammen grafisch dar (D).

Unterrichtsvorhaben Q1-II:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse <code>Queue</code></p> <ol style="list-style-type: none"> 1. Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen 2. Erarbeitung der Funktionalität der Klasse <code>Queue</code> 3. Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse <code>Queue</code> 	<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), • 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 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>2. Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse <code>Stack</code></p> <ol style="list-style-type: none"> 1. Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen 2. Erarbeitung der Funktionalität der Klasse <code>Stack</code> 3. Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse <code>Stack</code> 	
<p>3. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse <code>List</code></p> <ol style="list-style-type: none"> 1. Erarbeitung der Vorteile der Klasse <code>List</code> im Gegensatz zu den bereits bekannten linearen Strukturen 2. Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse <code>List</code> 	
<p>4. Vertiefung - Anwendungen von Listen, Stapeln oder Schlangen in mindestens einem weiteren Kontext</p>	

Unterrichtsvorhaben Q1-III:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Suchen von Daten in Listen und Arrays</p> <ol style="list-style-type: none">1. Lineare Suche in Listen und in Arrays2. Binäre Suche in Arrays als Beispiel für rekursives Problemlösen3. Untersuchung der beiden Suchverfahren hinsichtlich ihrer Effizienz (Laufzeitverhalten, Speicherbedarf) <p>2. Sortieren in Listen und Arrays - Entwicklung und Implementierung von iterativen und rekursiven Sortierverfahren</p> <ol style="list-style-type: none">1. Entwicklung und Implementierung eines einfachen Sortierverfahrens für eine Liste2. Implementierung eines einfachen Sortierverfahrens für ein Feld3. Entwicklung eines rekursiven Sortierverfahrens für ein Feld (z.B. Sortieren durch Mischen) <p>3. Untersuchung der Effizienz der Sortierverfahren „Sortieren durch direktes Einfügen“ und „Quicksort“ auf linearen Listen</p> <ol style="list-style-type: none">1. Grafische Veranschaulichung der Sortierverfahren2. Untersuchung der Anzahl der Vergleichsoperationen und des Speicherbedarfs bei beiden Sortierverfahren3. Beurteilung der Effizienz der beiden Sortierverfahren	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">• analysieren und erläutern Algorithmen und Programme (A),• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),• beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A),• entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M),• modifizieren Algorithmen und Programme (I),• implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),• implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I),• nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),• testen Programme systematisch anhand von Beispielen (I),• stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).

Unterrichtsvorhaben Q1-IV:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Nutzung von relationalen Datenbanken</p> <ol style="list-style-type: none"> 1. Aufbau von Datenbanken und Grundbegriffe <ol style="list-style-type: none"> 1. Entwicklung von Fragestellungen zur vorhandenen Datenbank 2. Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, Datenbankschema 2. SQL-Abfragen <ol style="list-style-type: none"> 1. Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL (SELECT (DISTINCT) ...FROM, WHERE, AND, OR, NOT) auf einer Tabelle 2. 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) 3. Vertiefung an einem weiteren Datenbankbeispiel 	<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),
<p>2. Modellierung von relationalen Datenbanken</p> <ol style="list-style-type: none"> 1. Entity-Relationship-Diagramm <ol style="list-style-type: none"> 1. Ermittlung von Entitäten, zugehörigen Attributen, Relationen und Kardinalitäten in Anwendungssituationen und Modellierung eines 	

<p>Datenbankentwurfs in Form eines Entity-Relationship-Diagramms</p> <ol style="list-style-type: none"> 2. Erläuterung und Modifizierung einer Datenbankmodellierung <p>2. Entwicklung einer Datenbank aus einem Datenbankentwurf</p> <ol style="list-style-type: none"> 1. Modellierung eines relationalen Datenbankschemas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln <p>3. Redundanz, Konsistenz und Normalformen</p> <ol style="list-style-type: none"> 1. Untersuchung einer Datenbank hinsichtlich Konsistenz und Redundanz in einer Anwendungssituation 2. 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"> • stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D), • überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D).
---	---

Unterrichtsvorhaben Q1-V:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Daten in Netzwerken und Sicherheitsaspekte in Netzen sowie beim Zugriff auf Datenbanken</p> <ol style="list-style-type: none">1. Beschreibung eines Datenbankzugriffs im Netz anhand eines Anwendungskontextes und einer Client-Server-Struktur zur Klärung der Funktionsweise eines Datenbankzugriffs2. Netztopologien als Grundlage von Client-Server-Strukturen und TCP/IP-Schichtenmodell als Beispiel für eine Paketübermittlung in einem Netz3. Vertraulichkeit, Integrität, Authentizität in Netzwerken sowie symmetrische und asymmetrische kryptografische Verfahren (Cäsar-, Vigenère-, RSA-Verfahren) als Methoden um Daten im Netz verschlüsselt zu übertragen	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">• beschreiben und erläutern Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A),• analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A),• untersuchen und bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen, die Sicherheit von Informatiksystemen sowie die Einhaltung der Datenschutzbestimmungen und des Urheberrechts (A),• untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A),• nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D).
<p>2. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht</p>	

Unterrichtsvorhaben Q2-I:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Analyse von Baumstrukturen in verschiedenen Kontexten</p> <ol style="list-style-type: none"> 1. Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit) 2. Aufbau und Darstellung von binären Bäumen anhand von Baumstrukturen in verschiedenen Kontexten 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), • analysieren und erläutern Algorithmen und Programme (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • modifizieren Algorithmen und Programme (I), • nutzen die Syntax und Semantik bei der Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I),
<p>2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse <code>BinaryTree</code></p> <ol style="list-style-type: none"> 1. Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext 2. Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms 3. Erarbeitung der Klasse <code>BinaryTree</code> und beispielhafte Anwendung der Operationen 4. Implementierung der Anwendung oder von Teilen der Anwendung 5. Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf 	
<p>3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse <code>BinarySearchTree</code></p> <ol style="list-style-type: none"> 1. Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen 2. Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramm, 3. grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften 	

<p>4. Erarbeitung der Klasse BinarySearchTree und Einführung des Interface Item zur Realisierung einer geeigneten Ordnungsrelation</p> <p>5. Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums</p>	<ul style="list-style-type: none"> • 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>4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen</p>	

Unterrichtsvorhaben Q2-II:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
1. Endliche Automaten <ol style="list-style-type: none">1. vom Automaten in den Schülerinnen und Schülern bekannten Kontexten zur formalen Beschreibung eines endlichen Automaten2. Untersuchung, Darstellung und Entwicklung endlicher Automaten	Die Schülerinnen und Schüler <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),
2. Untersuchung und Entwicklung von Grammatiken regulärer Sprachen <ol style="list-style-type: none">1. Erarbeitung der formalen Darstellung regulärer Grammatiken2. Untersuchung, Modifikation und Entwicklung von Grammatiken3. Entwicklung von endlichen Automaten zum Erkennen regulärer Sprachen, die durch Grammatiken gegeben werden4. Entwicklung regulärer Grammatiken zu endlichen Automaten	<ul style="list-style-type: none">• zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A),• ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A),• entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M),• entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M),• entwickeln zur akzeptierten Sprache eines Automaten die zugehörige Grammatik (M),• entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M),• modifizieren Grammatiken regulärer Sprachen (M),• entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M),
3. Grenzen endlicher Automaten	<ul style="list-style-type: none">• 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).• beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D).

Unterrichtsvorhaben Q2-III:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Von-Neumann-Architektur und die Ausführung maschinennaher Programme</p> <ol style="list-style-type: none">1. prinzipieller Aufbau einer von-Neumann-Architektur mit CPU, Rechenwerk, Steuerwerk, Register und Hauptspeicher2. maschinennahe Befehle und ihre Repräsentation in einem Binär-Code, der in einem Register gespeichert werden kann3. Analyse und Erläuterung der Funktionsweise eines einfachen maschinennahen Programms	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">• erläutern die Ausführung eines einfachen maschinennahen Programms sowie die Datenspeicherung auf einer „Von-Neumann-Architektur“ (A),• untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A).
<p>2. Grenzen der Automatisierbarkeit</p> <ol style="list-style-type: none">1. Vorstellung des Halteproblems2. Unlösbarkeit des Halteproblems3. Beurteilung des Einsatzes von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen	

Unterrichtsvorhaben Q2-IV:

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