

# **Schulinterner Lehrplan zum Kernlehrplan für die gymnasiale Oberstufe**

## **Informatik**

## **1 Die Fachgruppe Informatik des Carolus Magnus Gymnasium**

Das Fach Informatik wird am CMG ab der Jahrgangsstufe 8 im Wahlpflichtbereich II (WP II) dreistündig unterrichtet und von etwa einem Drittel der Schülerinnen und Schüler (SuS) besucht. In der zweijährigen Laufzeit dieser Kurse wird in altersstufengerechter Weise unter anderem auf den systematischen Umgang mit Daten und Informationen in den Office-Anwendungen, Grundlagen der Algorithmik am Beispiel einer didaktischen Lernumgebung (Scratch) und das Implementieren von Webseiten eingegangen.

In der Sekundarstufe II bietet das Carolus-Magnus-Gymnasium für die SuS Informatik als Grundkurs an. Ab dem Schuljahr 2015/16 wird bei entsprechenden Schülerwahlen ein Leistungskurs eingerichtet.

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

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

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

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

Informatik in der Sekundarstufe II wird momentan von zwei Lehrkräften unterrichtet. Die Arbeitsplätze in den zwei Computerräumen sind an das schulinterne Rechnernetz angeschlossen, so dass Schülerinnen und Schüler über einen individuell gestaltbaren Zugang zum zentralen Server der Schule alle Arbeitsplätze der drei Räume zum Zugriff auf ihre eigenen Daten, zur Recherche im Internet oder zur Bearbeitung schulischer Aufgaben verwenden können.

Der Unterricht erfolgt im 45-Minuten-Takt. Die Kursblockung sieht grundsätzlich für Grundkurse eine Doppelstunde und eine Einzelstunde vor.

## **2 Entscheidungen zum Unterricht**

### **2.1 Unterrichtsvorhaben**

Die Darstellung der Unterrichtsvorhaben im schulinternen Lehrplan besitzt den Anspruch, sämtliche im Kernlehrplan angeführten Kompetenzen abzudecken. Dies entspricht der Verpflichtung jeder Lehrkraft, Schülerinnen und Schülern Lerngelegenheiten zu ermögli-

chen, so dass alle Kompetenzerwartungen des Kernlehrplans von ihnen erfüllt werden können.

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

Im „Übersichtsraster Unterrichtsvorhaben“ (Kapitel 2.1.1) wird die für alle Lehrerinnen und Lehrer gemäß Fachkonferenzbeschluss verbindliche Verteilung der Unterrichtsvorhaben dargestellt. Das Übersichtsraster dient dazu, den Kolleginnen und Kollegen einen schnellen Überblick über die Zuordnung der Unterrichtsvorhaben zu den einzelnen Jahrgangsstufen sowie den im Kernlehrplan genannten Kompetenzen, Inhaltsfeldern und inhaltlichen Schwerpunkten zu verschaffen. Der ausgewiesene Zeitbedarf versteht sich als grobe Orientierungsgröße, die nach Bedarf über- oder unterschritten werden kann. Um Freiraum für Vertiefungen, besondere Schülerinteressen, aktuelle Themen bzw. die Erfordernisse anderer besonderer Ereignisse (z.B. Praktika, Kursfahrten o.ä.) zu erhalten, wurden im Rahmen dieses schulinternen Lehrplans ca. 75 Prozent der Bruttounterrichtszeit verplant.

Während der Fachkonferenzbeschluss zum „Übersichtsraster Unterrichtsvorhaben“ zur Gewährleistung vergleichbarer Standards sowie zur Absicherung von Lerngruppenübertritten und Lehrkraftwechseln für alle Mitglieder der Fachkonferenz Bindekraft entfalten soll, beinhaltet die Ausweisung „konkretisierter Unterrichtsvorhaben“ (Kapitel 2.1.2) Beispiele und Materialien, die empfehlenden Charakter haben. Referendarinnen und Referendaren sowie neuen Kolleginnen und Kollegen dienen diese vor allem zur standardbezogenen Orientierung in der neuen Schule, aber auch zur Verdeutlichung von unterrichtsbezogenen fachgruppeninternen Absprachen zu didaktisch-methodischen Zugängen, fächerübergreifenden Kooperationen, Lernmitteln und -orten sowie vorgesehenen Leistungsüberprüfungen, die im Einzelnen auch den Kapiteln 2.2 bis 2.3 zu entnehmen sind.

Da in den folgenden Unterrichtsvorhaben Inhalte in der Regel anhand von Problemstellungen in Anwendungskontexten bearbeitet werden, werden in einigen Unterrichtsvorhaben jeweils mehrere Inhaltsfelder angesprochen.

## 2.1.1 Übersichtsraster Unterrichtsvorhaben

### I) Einführungsphase

Einführungsphase	
<p><u>Unterrichtsvorhaben E-I</u></p> <p><b>Thema:</b> <i>Einführung in die Nutzung von Informatiksystemen und grundlegende Begrifflichkeiten</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"><li>• Argumentieren</li><li>• Darstellen und Interpretieren</li><li>• Kommunizieren und Kooperieren</li></ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"><li>• Informatiksysteme</li><li>• Informatik, Mensch und Gesellschaft</li></ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"><li>• Einzelrechner</li><li>• Dateisystem</li><li>• Einsatz von Informatiksystemen</li></ul> <p><b>Zeitbedarf:</b> 8 Stunden</p>	<p><u>Unterrichtsvorhaben E-II</u></p> <p><b>Thema:</b> <i>Grundlagen der OOP und erste Erstellung von animierten Welten in Greenfoot</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"><li>• Argumentieren</li><li>• Modellieren</li><li>• Implementieren</li><li>• Darstellen und Interpretieren</li></ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"><li>• Daten und ihre Strukturierung</li><li>• Syntax und Semantik einer Programmiersprache</li><li>• Formale Sprachen und Automaten</li></ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"><li>• Objekte und Klassen</li><li>• Syntax und Semantik einer Programmiersprache</li></ul> <p><b>Zeitbedarf:</b> 8 Stunden</p>

## Einführungsphase

### Unterrichtsvorhaben E-III

**Thema:**

*Nutzung von Kontrollstrukturen und  
Struktogrammen zur Problemlösung in Greenfoot*

**Zentrale Kompetenzen:**

- Implementieren
- Darstellen und Interpretieren

**Inhaltsfelder:**

- Syntax und Semantik einer Programmiersprache
- Formale Sprachen und Automaten

**Inhaltliche Schwerpunkte:**

- Syntax und Semantik einer Programmiersprache

**Zeitbedarf:** 7 Stunden

### Unterrichtsvorhaben E-IV

**Thema:**

*Problemlösestrategien*

**Zentrale Kompetenzen:**

- Argumentieren
- Modellieren
- Implementieren
- Kommunizieren und Kooperieren

**Inhaltsfelder:**

- Algorithmen
- Formale Sprachen und Automaten

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 11 Stunden

## Einführungsphase

### Unterrichtsvorhaben E-V

**Thema:**

*Modellierung und Implementierung von Klassen mit Attributen und Methoden*

**Zentrale Kompetenzen:**

- 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

**Zeitbedarf:** 11 Stunden

### Unterrichtsvorhaben E-VI

**Thema:**

*Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand der Spieleentwicklung in Greenfoot*

**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

**Zeitbedarf:** 12 Stunden

<b>Einführungsphase</b>	
<p><u>Unterrichtsvorhaben E-VII</u></p> <p><b>Thema:</b> <i>Such- und Sortieralgorithmen anhand kontextbezogener Beispiele</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>• Argumentieren</li> <li>• Modellieren</li> <li>• Darstellen und Interpretieren</li> <li>• Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>• Algorithmen</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>• Algorithmen zum Suchen und Sortieren</li> <li>• Analyse, Entwurf und Implementierung einfacher Algorithmen</li> </ul> <p><b>Zeitbedarf:</b> 9 Stunden</p>	<p><u>Unterrichtsvorhaben E-VIII</u></p> <p><b>Thema:</b> <i>Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>• Argumentieren</li> <li>• Darstellen und Interpretieren</li> <li>• Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>• Informatik, Mensch und Gesellschaft</li> <li>• Informatiksysteme</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>• Wirkungen der Automatisierung</li> <li>• Geschichte der automatischen Datenverarbeitung</li> <li>• Digitalisierung</li> </ul> <p><b>Zeitbedarf:</b> 10 Stunden</p>
<b>Summe Einführungsphase: 76</b>	

### **2.1.2 Konkretisierte Unterrichtsvorhaben**

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

In der Einführungsphase wird die didaktische Entwicklungsumgebung Greenfoot verwendet. Installationspakete finden sich unter:

<http://www.greenfoot.org/download> (abgerufen am 20.09.2014)

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

<http://www.standardsicherung.schulministerium.nrw.de/abitur-gost/fach.php?fach=15> (abgerufen: 30. 04. 2014)



## I) Einführungsphase

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

Die Schülerinnen und Schüler

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

## Unterrichtsvorhaben EF-I

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

**Leitfragen:** *Wie werden Dateien und Ordner sinnvoll strukturiert? Wie nutzt man ein Cloud-System (Dropbox) effizient und verantwortungsbewusst? Was ist die von Neumann-Architektur? Wie werden Daten im Rechner gespeichert.*

### **Vorhabenbezogene Konkretisierung:**

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

Zunächst lernen die SuS das schuleigene Netzwerk zur Arbeit in der Schule und das Cloud-System Dropbox zum Transfer der Daten kennen. Anschließend 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.

Desweiteren lernen die SuS das Binärsystem zur Speicherung von Informationen auf einem Rechner kennen.

Zeitbedarf: 8 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><b>1. Speicherung von Informationen</b></p> <p>(a) Informatik als Wissenschaft der Verarbeitung von Informationen</p> <p>(b) Speichern von Daten mit informatischen Systemen am Beispiel der Schulrechner und der Dropbox</p> <p>(c) Vereinbarung von Richtlinien zur Datenspeicherung auf den Schulrechnern (z.B. Ordnerstruktur, Dateibezeichner usw.)</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• beschreiben und erläutern den Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A),</li> <li>• nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D),</li> <li>• stellen ganze Zahlen und Zeichen in Binärcodes dar (D),</li> <li>• interpretieren Binärcodes als Zahlen und Zeichen (D),</li> <li>• nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K).</li> </ul>	<p><i>Material:</i> Video über „von Neumann Architektur“</p> <p><i>Beispiel:</i> Textkodierung Kodierung und Dekodierung von Texten mit Hilfe des ASCII-Codes</p>
<p><b>2. Aufbau informatischer Systeme</b></p> <p>(a) Kennenlernen der „Von-Neumann-Architektur“</p> <p>(b) Identifikation typischer Komponenten informatischer Systeme und Einordnung in die von Neumann Architektur</p>		
<p><b>3. Kodierung von Informationen</b></p> <p>(a) Darstellung von Informationen in Schrift, Bild und Ton</p> <p>(b) Darstellung von Informationen im Binärsystem</p>		

## Unterrichtsvorhaben EF-II

**Thema:** Grundlagen der OOP und erste Erstellung von animierten Welten in Greenfoot

**Leitfrage:** Wie lassen sich Gegenstände im Sinne der OOP modellieren? Wie arbeitet man mit Greenfoot?

### Vorhabenbezogene Konkretisierung:

Die zentralen Grundlagen der OOP werden eingeführt.

In EF–II bis EF-VI wird mit der Unterrichtsreihe [http://bildungsserver.berlin-brandenburg.de/inf\\_greenfoot1.html](http://bildungsserver.berlin-brandenburg.de/inf_greenfoot1.html) gearbeitet.

**Zeitbedarf:** 8 Stunden

### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<b>1. Kennenlernen von Greenfoot</b>	Die Schülerinnen und Schüler <ul style="list-style-type: none"><li>• ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li><li>• modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M),</li><li>• implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I)</li><li>• modellieren Klassen unter Verwendung von Vererbung (M),</li><li>• stellen den Zustand eines Objekts dar (D).</li></ul>	Unterrichtsreihe auf dem Bildungsserver Berlin-Brandenburg
<b>2. Identifikation von Objekten</b> <ul style="list-style-type: none"><li>(a) An einem realen Beispiel werden Objekte im Sinne der OO-Modellierung eingeführt.</li><li>(b) Typengleiche Objekte werden zu einer Klassenzusammengefasst.</li><li>(c) Vertiefung: Modellierung weiterer Beispiele ähnlichen Musters</li></ul>		
<b>3. Implementierung erster Animationen</b> <ul style="list-style-type: none"><li>(a) Grundaufbau einer Java-Klasse</li><li>(b) Nutzung von Methoden zur Bewegung von Objekten</li></ul>		

## Unterrichtsvorhaben EF-III

**Thema:** Nutzung von Kontrollstrukturen und Struktogrammen zur Problemlösung in Greenfoot

**Leitfrage:** Wie können Objekte mit Hilfe von Schleifen und bedingten Anweisungen effektiv gesteuert werden? Wie können algorithmische Grundstrukturen mit Struktogrammen modelliert werden?

### Vorhabenbezogene Konkretisierung:

Die Problemstellungen in Greenfoot werden komplexer. Kontrollstrukturen werden eingesetzt, um die Probleme effizient zu lösen.

**Zeitbedarf:** 7 Stunden

### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<b>1. Kontrollstrukturen nutzen</b> (a) Schleifen (while und do-while) werden genutzt um Objekte effizient und flexibel durch die Greenfoot-Welt zu bewegen (b) Bedingte Anweisungen werden eingeführt, damit Objekte flexibel auf andere Objekte reagieren können	Die Schülerinnen und Schüler <ul style="list-style-type: none"><li>• implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),</li><li>• implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),</li><li>• analysieren und erläutern einfache Algorithmen und Programme (A).</li></ul>	Unterrichtsreihe auf dem Bildungsserver Berlin-Brandenburg
<b>2. Struktogramme zur Modellierung algorithmischer Grundstrukturen</b> (a) Die algorithmischen Grundstrukturen, die mit Hilfe der Kontrollstrukturen implementiert werden durch Struktogramme modelliert. Die in den Struktogrammen modellierten Algorithmen werden implementiert.		

## Unterrichtsvorhaben EF-IV

**Thema:** Problemlösestrategien

**Leitfrage:** Was sind die Varianten/Invarianten eines Problems? Wie kann man Probleme mit Hilfe der Top-Down-Strategie lösen, wie helfen Struktogramme dabei.

### Vorhabenbezogene Konkretisierung:

In einer schwerpunktmäßigen Theoriesequenz werden die zentralen Problemlösestrategien eingeführt.

**Zeitbedarf:** 11 Stunden

### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<b>1. Problemlösenstrategien</b> (a) Herausstellen von Varianten und Invarianten eines Problems (b) Anwenden der Top-Down-Strategie zur Problemanalyse (c) Entwickeln komplexerer Struktogramme zur Problemlösung und Implementierung dieser Diagramme (d) Nutzen diese Teilstrategien zur Lösung des Gesamtproblems	Die Schülerinnen und Schüler <ul style="list-style-type: none"><li>• analysieren und erläutern einfache Algorithmen und Programme(A),</li><li>• entwerfen eigene Algorithmen und stellen sie umgangssprachlich und grafisch dar (M),</li><li>• implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),</li><li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I).</li></ul>	Unterrichtsreihe auf dem Bildungsserver Berlin-Brandenburg

## Unterrichtsvorhaben EF-V

**Thema:** Modellierung und Implementierung von Klassen mit Attributen und Methoden

**Leitfrage:** Wie lassen sich Klassen vollständig implementieren? Wie helfen die erlernten Kontrollstrukturen dabei?

### Vorhabenbezogene Konkretisierung:

Alle bisher vermittelten Kompetenzen werden genutzt um vollständige Klassen mit komplexeren Methoden zu implementieren.

**Zeitbedarf:** 11 Stunden

### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<b>1. Variablen und Schleifen</b> (a) int-Variablen (lokale Variablen) werden zum Zählen genutzt (b) Nutzen die for-Schleife und vergleichen mit der while-Schleife (c) Verwenden und implementieren Methoden mit Parametern und Rückgabewerten	Die Schülerinnen und Schüler <ul style="list-style-type: none"><li>ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li><li>modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M),</li><li>ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M),</li><li>implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I).</li></ul>	Unterrichtsreihe auf dem Bildungsserver Berlin-Brandenburg
<b>2. Klassen</b> (a) Implementieren von Attributen, Konstruktoren und Standardmethoden (b) Implementierung eines Klassendiagramms (c) Dokumentieren der Klassen im Javadoc-Format.		

## Unterrichtsvorhaben EF-VI

**Thema:** Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand der Spieleentwicklung in Greenfoot

**Leitfrage:** Wie lassen sich Gegenstände im Sinne der OOP modellieren? Wie arbeitet man mit Greenfoot?

### Vorhabenbezogene Konkretisierung:

Zunächst wird ein eigenes Projekt unter Anleitung entwickelt, anschließend entwickeln die SuS ein eigenes Projekt. Dabei halten sie sich jedoch an den Workflow des gemeinsam entwickelten Projektes.

**Zeitbedarf:** 12 Stunden

### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<b>1. Implementierung eines Projektes nach Anleitung</b>	Die Schülerinnen und Schüler	Unterrichtsreihe auf dem Bildungsserver Berlin-Brandenburg
<b>2. Selbständige Implementierung eines Projektes</b>	<ul style="list-style-type: none"><li>• analysieren und erläutern eine OO-Modellierung (A),</li><li>• modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M),</li><li>• stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D),</li><li>• implementieren Klassen unter Nutzung dokumentierter Klassenbibliotheken,</li><li>• testen Programme schrittweise anhand von Beispielen (I).</li></ul>	Arbeitsmaterial zur Top-Down - Projektentwicklung

## Unterrichtsvorhaben EF-VII (aus Beispiel übernommen, wird erprobt und angepasst)

**Thema:** Such- und Sortieralgorithmen anhand kontextbezogener Beispiele

**Leitfragen:** *Wie können Objekte bzw. Daten effizient sortiert werden, so dass eine schnelle Suche möglich wird?*

### Vorhabenbezogene Konkretisierung:

Dieses Unterrichtsvorhaben beschäftigt sich mit der 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.

**Zeitbedarf:** 9 Stunden

### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<b>1. Explorative Erarbeitung eines Sortierverfahrens</b> (a) Sortierprobleme im Kontext informatischer Systeme und im Alltag (z.B. Dateisortierung, Tabellenkalkulation, Telefonbuch, Bundesligatabelle, usw.) (b) Vergleich zweier Elemente als Grundlage eines Sortieralgorithmus	Die Schülerinnen und Schüler <ul style="list-style-type: none"><li>• beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeit und Speicherplatzbedarf (A),</li><li>• entwerfen einen weiteren Algorithmus zum Sortieren (M),</li><li>• analysieren Such- und Sortieralgorithmen</li></ul>	<i>Beispiel:</i> Sortieren mit Waage Die Schülerinnen und Schüler bekommen die Aufgabe, kleine, optisch identische Kunststoffbehälter aufsteigend nach ihrem Gewicht zu sortieren. Dazu steht ihnen eine Balkenwaage zur Verfügung, mit deren Hilfe sie das Gewicht zweier Behälter vergleichen können.



<p>(c) Erarbeitung eines Sortieralgorithmus durch die Schülerinnen und Schüler</p>	<p>und wenden sie auf Beispiele an (D).</p>	<p><i>Materialien:</i> Computer science unplugged – Sorting Algorithms, URL: <a href="http://www.csunplugged.org/sorting-algorithms">www.csunplugged.org/sorting-algorithms</a> abgerufen: 30. 03. 2014</p>
<p><b>2. Systematisierung von Algorithmen und Effizienzbetrachtungen</b></p> <p>(a) Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode (auf jeden Fall: Sortieren durch Vertauschen, Sortieren durch Auswählen)</p> <p>(b) Anwendung von Sortieralgorithmen auf verschiedene Beispiele</p> <p>(c) Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche</p> <p>(d) Variante des Sortierens durch Auswählen (Nutzung eines einzigen oder zweier Felder bzw. lediglich eines einzigen zusätzlichen Ablageplatzes oder mehrerer neuer Ablageplätze)</p> <p>(e) Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des Speicherplatzbedarfs</p> <p>(f) Analyse des weiteren Sortieralgorithmus (sofern nicht in Sequenz 1 und 2 bereits geschehen)</p>		<p><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. Kenntnisse in rekursiver Programmierung sind nicht erforderlich, da eine Implementierung nicht angestrebt wird.</p> <p><i>Materialien:</i> Computer science unplugged – Sorting Algorithms, URL: <a href="http://www.csunplugged.org/sorting-algorithms">www.csunplugged.org/sorting-algorithms</a> abgerufen: 30. 03. 2014</p>
<p><b>3. Binäre Suche auf sortierten Daten</b></p> <p>(a) Suchaufgaben im Alltag und im Kontext informatischer Systeme</p>		<p><i>Beispiel:</i> Simulationsspiel zur binären Suche nach Tischtennisbällen Mehrere Tischtennisbälle sind nummeriert, sortiert</p>

<p>(b) Evtl. Simulationsspiel zum effizienten Suchen mit binärer Suche</p> <p>(c) Effizienzbetrachtungen zur binären Suche</p>		<p>und unter Bechern verdeckt. Mit Hilfe der binären Suche kann sehr schnell ein bestimmter Tischtennisball gefunden werden.</p> <p><i>Materialien:</i>  Computer science unplugged – Searching Algorithms, URL:  <a href="http://www.csunplugged.org/searching-algorithms">www.csunplugged.org/searching-algorithms</a>, abgerufen: 30. 03. 2014</p>
--	--	---

#### Unterrichtsvorhaben EF-VIII (aus Beispiel übernommen, wird erprobt und angepasst)

**Thema:** Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes

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

#### **Vorhabenbezogene Konkretisierung:**

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

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

**Zeitbedarf:** 10 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><b>1. Selbstständige Erarbeitung von Themen durch die Schülerinnen und Schüler</b></p> <p>(a) Mögliche Themen zur Erarbeitung in Kleingruppen:</p> <ul style="list-style-type: none"> <li>• „Eine kleine Geschichte der Digitalisierung: vom Morsen zum modernen Digitalcomputer“</li> <li>• „Eine kleine Geschichte der Kryptographie: von Caesar zur Enigma“</li> <li>• „Von Nullen, Einsen und mehr: Stellenwertsysteme und wie man mit ihnen rechnet“</li> <li>• „Kodieren von Texten und Bildern: ASCII, RGB und mehr“</li> <li>• „Auswirkungen der Digitalisierung: Veränderungen der Arbeitswelt und Datenschutz“</li> </ul> <p>(b) Vorstellung durch SuS</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A),</li> <li>• erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A),</li> <li>• stellen ganze Zahlen und Zeichen in Binärcodes dar (D),</li> <li>• interpretieren Binärcodes als Zahlen und Zeichen (D),</li> <li>• nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation. (K).</li> </ul>	<p><i>Beispiel:</i> Ausstellung zu informatischen Themen Die Schülerinnen und Schüler bereiten eine Ausstellung zu informatischen Themen vor. Dazu werden Stellwände und Plakate vorbereitet, die ggf. auch außerhalb des Informatikunterrichts in der Schule ausgestellt werden können.</p> <p><i>Materialien:</i> Schülerinnen und Schüler recherchieren selbstständig im Internet, in der Schulbibliothek, in öffentlichen Bibliotheken, usw.</p>
<p><b>2. Vertiefung des Themas Datenschutz</b></p> <p>(a) Erarbeitung grundlegender Begriffe des Datenschutzes</p> <p>(b) Problematisierung und Anknüpfung an die Lebenswelt der SuS</p> <p>(c) Diskussion von Fallbeispielen aus dem Themenbereich „Datenschutz“</p>		<p><i>Beispiel:</i> Die SuS bearbeiten Fallbeispiele aus ihrer eigenen Erfahrungswelt</p> <p><i>Materialien:</i> Materialblatt zum Bundesdatenschutzgesetz (<a href="#">Download EF-VI.1</a>)</p>

## II) Qualifikationsphase (Q1 und Q2) – GRUNKURS/LEISTUNGSKURS

Qualifikationsphase 1	
<p><u>Unterrichtsvorhaben Q1-I</u></p> <p><b>Thema:</b> Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>• Argumentieren</li> <li>• Modellieren</li> <li>• Implementieren</li> <li>• Darstellen und Interpretieren</li> <li>• Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>• Daten und ihre Strukturierung</li> <li>• Algorithmen</li> <li>• Formale Sprachen und Automaten</li> <li>• Informatiksysteme</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>• Objekte und Klassen</li> <li>• Analyse, Entwurf und Implementierung von Algorithmen</li> <li>• Syntax und Semantik einer Programmiersprache</li> <li>• Nutzung von Informatiksystemen</li> </ul> <p><b>Zeitbedarf:</b> 8 Stunden/13 Stunden</p>	<p><u>Unterrichtsvorhaben Q1-II</u></p> <p><b>Thema:</b> Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen <i>Implementierung der Datenstrukturen Queue, Stack, List</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>• Argumentieren</li> <li>• Modellieren</li> <li>• Implementieren</li> <li>• Darstellen und Interpretieren</li> <li>• Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>• Daten und ihre Strukturierung</li> <li>• Algorithmen</li> <li>• Formale Sprachen und Automaten</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>• Objekte und Klassen</li> <li>• Analyse, Entwurf und Implementierung von Algorithmen</li> <li>• Algorithmen in ausgewählten informatischen Kontexten</li> <li>• Syntax und Semantik einer Programmiersprache</li> </ul> <p><b>Zeitbedarf:</b> 20 Stunden /26 Stunden</p>

## 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

**Zeitbedarf:** 16 Stunden/25 Stunden

### Unterrichtsvorhaben Q1-IV

**Thema:**

*Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten, Datenbankzugriff über Java*

**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

**Zeitbedarf:** 20 Stunden/35 Stunden

## 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

**Zeitbedarf:** 10 Stunden/11 Stunden

**Summe Qualifikationsphase 1: 74 Stunden/110 Stunden**

## Qualifikationsphase 2

### Unterrichtsvorhaben Q2-I

**Thema:**

Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen  
Implementierung von BinaryTree, Binary-SearchTree, Graph, Vertex und Edge, Interfaces

**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

**Zeitbedarf:** 24 Stunden /35 Stunden

### Unterrichtsvorhaben Q2-II

**Thema:**

Endliche Automaten und reguläre Sprachen  
Kellerautomaten und kontextfreie Grammatiken  
Scanner, Parser, Interpreter

**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

**Zeitbedarf:** 20 Stunden /40 Stunden

## Qualifikationsphase 2

### Unterrichtsvorhaben Q2-III

**Thema:**

*Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit*

**Zentrale Kompetenzen:**

- Argumentieren
- Kommunizieren und Kooperieren

**Inhaltsfelder:**

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

**Inhaltliche Schwerpunkte:**

- Einzelrechner und Rechnernetzwerke
- Grenzen der Automatisierung

**Zeitbedarf:** 12 Stunden/15 Stunden

**Summe Qualifikationsphase 2: 56 Stunden/90 Stunden**



---

## Übersichtsraster Unterrichtsvorhaben

### Qualifikationsphase (Grundkurs/Leistungskurs)

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

Die Schülerinnen und Schüler

- verwenden die Fachsprache bei der Kommunikation über informatische Sachverhalte (K),
- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung von Dateien unter Berücksichtigung der Rechteverwaltung (K),
- organisieren und koordinieren kooperatives und eigenverantwortliches Arbeiten (K),
- strukturieren den Arbeitsprozess, vereinbaren Schnittstellen und führen Ergebnisse zusammen (K),
- beurteilen Arbeitsorganisation, Arbeitsabläufe und Ergebnisse (K),
- präsentieren Arbeitsabläufe und -ergebnisse adressatengerecht (K).

### Unterrichtsvorhaben Q1-I:

#### Thema: Wiederholung der objektorientierten Modellierung und Programmierung

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

#### Vorhabenbezogenen Konkretisierung:

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

Die Schülerinnen und Schülern erläutern und modifizieren den ersten Entwurf und modellieren sowie implementieren weitere Klassen und Methoden für eine entsprechende Anwendung. Klassen und ihre Beziehungen werden in einem Implementationsdiagramm dargestellt. Dabei werden Sichtbarkeitsbereiche zugeordnet. Exemplarisch wird eine Klasse dokumentiert. Der Nachrichtenaustausch zwischen verschiedenen Objekten wird verdeutlicht, indem die Kommunikation zwischen zwei ausgewählten Objekten grafisch dargestellt wird. In diesem Zusammenhang wird das Nachrichtenkonzept der objektorientierten Programmierung wiederholt.

**Zeitbedarf:** 8 Stunden(GK) / 13 Stunden (LK)

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><b>Wiederholung und Erweiterung der objektorientierten Modellierung und Programmierung durch Analyse und Erweiterung eines kontextbezogenen Beispiels</b></p> <ol style="list-style-type: none"> <li>1. Analyse der Problemstellung</li> <li>2. Analyse der Modellierung (Implementationsdiagramm)</li> <li>3. Erweiterung der Modellierung im Implementationsdiagramm (Vererbung, abstrakte Klasse)</li> <li>4. Kommunikation zwischen mindestens zwei Objekten (grafische Darstellung)</li> <li>5. Dokumentation von Klassen</li> <li>6. Nutzung dokumentierter Klassenbibliotheken</li> </ol>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (<b>M</b>)</li> <li>• stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (<b>D</b>)</li> <li>• modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (<b>M</b>)</li> <li>• modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (<b>M</b>)</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (<b>M</b>)</li> <li>• verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie (<b>M</b>)</li> <li>• ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (<b>M</b>)</li> <li>• stellen die Kommunikation zwischen Objekten graphisch dar (<b>D</b>)</li> <li>• stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (<b>D</b>),</li> <li>• dokumentieren Klassen (<b>D</b>)</li> <li>• analysieren und erläutern objektorientierte Modellierungen (<b>A</b>)</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (<b>I</b>)</li> </ul>	<p>Beispiel: Online-Rollenspiel</p> <p>In einem Online-Rollenspiel kämpfen Helden und Monster gegeneinander. Dabei werden Zauberer und Krieger als Unterklassen von der Klasse Held abgeleitet.</p> <p>(Material: Informatik 2, Schöningh)</p>

---

## Unterrichtsvorhaben Q1-II:

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

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

#### **Vorhabenbezogene Konkretisierung**

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

**Zeitbedarf:** 20 Stunden(GK) / 26 Stunden (LK)

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><b>1. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse <code>Queue</code></b></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse <code>Queue</code></p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse <code>Queue</code></p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A),</li> <li>• analysieren und erläutern Algorithmen und Programme (A),</li> <li>• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M),</li> <li>• ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li> <li>• modifizieren Algorithmen und Programme (I),</li> <li>• implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),</li> <li>• nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>• testen Programme systematisch anhand von Beispielen (I),</li> <li>• stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D).</li> </ul>	<p><i>Beispiel:</i> Patientenwarteschlange (jeder kennt seinen Nachfolger bzw. alternativ: seinen Vorgänger) Sobald ein Patient in einer Arztpraxis eintrifft, werden sein Name und seine Krankenkasse erfasst. Die Verwaltung der Patientenwarteschlange geschieht über eine Klasse, die hier als Wartezimmer bezeichnet wird. Wesentliche Operationen sind das „Hinzufügen“ eines Patienten und das „Entfernen“ eines Patienten, wenn er zur Behandlung gerufen wird. Die Simulationsanwendung stellt eine GUI zur Verfügung, legt ein Wartezimmer an und steuert die Abläufe. Wesentlicher Aspekt des Projektes ist die Modellierung des Wartezimmers mit Hilfe der Klasse <code>Queue</code>. Anschließend wird der Funktionsumfang der Anwendung erweitert: Patienten können sich zusätzlich in die Warteschlange zum Blutdruckmessen einreihen. Objekte werden von zwei Schlangen verwaltet.</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.2 – Warteschlange (<a href="#">Download Q1-II.1</a>)</p>
<p><b>2. Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse <code>Stack</code></b></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse <code>Stack</code></p> <p>(c) Modellierung und Implementierung der An-</p>		<p><i>Beispiel:</i> Bücherstapel In einem Bücherstapel soll bestimmtes Buch gefunden werden</p> <p>oder</p> <p><i>Beispiel:</i> Türme von Hanoi</p>

<p>wendung unter Verwendung eines oder mehrerer Objekte der Klasse <code>Stack</code></p>		
<p><b>3. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse <code>List</code></b></p> <p>(a) Erarbeitung der Vorteile der Klasse <code>List</code> im Gegensatz zu den bereits bekannten linearen Strukturen</p> <p>(b) Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse <code>List</code>.</p>		<p><i>Beispiel:</i> Rangliste Abfahrtslauf</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.2 - Listen</p> <p>Oder</p> <p>Spendenlauf, Kurslisten</p>
<p><b>4. Vertiefung - Anwendungen von Listen, Stapeln oder Schlangen in mindestens einem weiteren Kontext</b></p>		<p><i>Beispiel:</i> Terme in Postfix-Notation <i>Beispiel:</i> Rangierbahnhof <i>Beispiel:</i> Autos an einer Ampel zur Zufahrtsregelung</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1-II.3 – Anwendungen für lineare Datenstrukturen</p>

## Unterrichtsvorhaben Q1-III:

**Thema:** Suchen und Sortieren auf linearen Datenstrukturen

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

### Vorhabenbezogene Konkretisierung:

In einem Anwendungskontext werden zunächst Informationen in einer linearen Liste bzw. einem Feld gesucht. Hierzu werden Verfahren entwickelt und implementiert bzw. analysiert und erläutert, wobei neben einem iterativen auch ein rekursives Verfahren thematisiert wird und mindestens ein Verfahren selbst entwickelt und implementiert wird. Die verschiedenen Verfahren werden hinsichtlich Speicherbedarf und Zahl der Vergleichsoperationen miteinander verglichen.

Anschließend werden Sortierverfahren entwickelt und implementiert (ebenfalls für lineare Listen und Felder). Hierbei soll auch ein rekursives Sortierverfahren entwickelt werden. Die Implementationen von Quicksort sowie dem Sortieren durch Einfügen werden analysiert und erläutert. Falls diese Verfahren vorher schon entdeckt wurden, sollen sie hier wiedererkannt werden. Die rekursive Abarbeitung eines Methodenaufrufs von Quicksort wird grafisch dargestellt.

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

**Zeitbedarf:** 16 Stunden (GK) / 25 Stunden (LK)

### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<b>1. Suchen von Daten in Listen und Arrays</b> (a) Lineare Suche in Listen und in Arrays (b) Binäre Suche in Arrays als Beispiel für rekursives Problemlösen (c) Untersuchung der beiden Suchverfahren hinsichtlich ihrer Effizienz (Laufzeitverhalten, Speicherbedarf)	Die Schülerinnen und Schüler <ul style="list-style-type: none"><li>analysieren und erläutern Algorithmen und Programme (A),</li><li>beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li><li>beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A),</li><li>entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M),</li><li>modifizieren Algorithmen und Programme (I),</li><li>implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),</li></ul>	<i>Beispiel:</i> Zahl in Liste suchen  <i>Beispiel:</i> Karteiverwaltung (Objekt suchen) Für ein Adressverwaltungsprogramm soll eine Methode zum Suchen einer Adresse geschrieben werden.  <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.3 - Suchen und Sortieren
<b>2. Sortieren in Listen und Arrays - Entwicklung und Implementierung von iterativen und rekursiven Sortierverfahren</b> (a) Entwicklung und Implementierung eines einfachen Sortierverfahrens für eine Liste (Select Sort) (b) Implementierung eines einfachen Sortierver-		<i>Siehe oben</i>

fahrens für ein Feld (c) Entwicklung eines rekursiven Sortierverfahren für ein Feld (z.B. Sortieren durch Mischen) (Merge oder Quick Sort)	<ul style="list-style-type: none"> <li>• implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I),</li> <li>• nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> </ul>	
<b>3. Untersuchung der Effizienz der Sortierverfahren „Sortieren durch Auswahl“ und „Quicksort/Mergesort“ auf linearen Listen</b> (a) Grafische Veranschaulichung der Sortierverfahren (b) Untersuchung der Anzahl der Vergleichsoperationen und des Speicherbedarf bei beiden Sortierverfahren (c) Beurteilung der Effizienz der beiden Sortierverfahren	<ul style="list-style-type: none"> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>• testen Programme systematisch anhand von Beispielen (I),</li> <li>• stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).</li> </ul>	<i>Beispiel:</i> Karteiverwaltung (s.o.)  oder <i>Beispiel:</i> Bundesjugendspiele (s.o.)  <i>Materialien:</i> (s.o.)

---

## **Unterrichtsvorhaben Q1-IV:**

**Thema:** Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

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

### **Vorhabenbezogene Konkretisierung:**

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

In anderen Anwendungskontexten müssen Datenbanken erst noch entwickelt werden, um Daten zu speichern und Informationen für die Beantwortung von möglicherweise auftretenden Fragen zur Verfügung zu stellen. Dafür ermitteln Schülerinnen und Schüler in den Anwendungssituationen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten und stellen diese in Entity-Relationship-Modellen dar. Entity-Relationship-Modelle werden interpretiert und erläutert, modifiziert und in Datenbankschemata überführt. Mit Hilfe von SQL-Anweisungen können anschließend im Kontext relevante Informationen aus der Datenbank extrahiert werden.

Ein Entity-Relationship-Diagramm kann auch verwendet werden, um die Entitäten inklusive ihrer Attribute und Relationen in einem vorgegebenen Datenbankschema darzustellen.

An einem Beispiel wird verdeutlicht, dass in Datenbanken Redundanzen unerwünscht sind und Konsistenz gewährleistet sein sollte. Die 1. bis 3. Normalform wird als Gütekriterium für Datenbankentwürfe eingeführt. Datenbankschemata werden hinsichtlich der 1. bis 3. Normalform untersucht und (soweit nötig) normalisiert.

**Zeitbedarf:** 20 Stunden (GK) / 35 Stunden (LK)



## Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><b>1. Nutzung von relationalen Datenbanken</b></p> <p>(a) Aufbau von Datenbanken und Grundbegriffe</p> <ul style="list-style-type: none"> <li>• Entwicklung von Fragestellungen zur vorhandenen Datenbank</li> <li>• Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, Datenbankschema</li> </ul> <p>(b) SQL-Abfragen</p> <ul style="list-style-type: none"> <li>• Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL (SELECT (DISTINCT) ...FROM, WHERE, AND, OR, NOT) auf einer Tabelle</li> <li>• 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: =, &lt;&gt;, &gt;, &lt;, &gt;=, &lt;=, LIKE, BETWEEN, IN, IS NULL)</li> </ul> <p>(c) Vertiefung an einem weiteren Datenbankbeispiel</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (<b>M</b>)</li> <li>• stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (<b>D</b>),</li> <li>• modifizieren eine Datenbankmodellierung (<b>M</b>),</li> <li>• modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (<b>M</b>),</li> <li>• bestimmen Primär- und Sekundärschlüssel (<b>M</b>),</li> <li>• analysieren und erläutern eine Datenbankmodellierung (<b>A</b>),</li> <li>• erläutern die Eigenschaften normalisierter Datenbankschemata (<b>A</b>)</li> <li>• überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (<b>D</b>)</li> <li>• überprüfen Datenbankschemata in die 1. Bis 3. Normalform (<b>M</b>)</li> </ul>	<p><i>Beispiele:</i> Kurs- oder Schulverwaltung</p> <p><i>Materialien:</i>  <i>Beispiel:</i> Video-Center  <a href="http://dokumentation.videocenter.schule.de/old/video/index.html">http://dokumentation.videocenter.schule.de/old/video/index.html</a>  ist die Simulation einer Online-Videothek für den Informatikunterricht</p> <p><i>Beispiel:</i> Schulbuchausleihe Unter  <a href="http://www.brd.nrw.de/lerntreffs/informatik/structure/material/sek2/datenbanken.php">www.brd.nrw.de/lerntreffs/informatik/structure/material/sek2/datenbanken.php</a> (abgerufen: 30. 03. 2014) wird eine Datenbank zur Verfügung gestellt, die Daten einer Schulbuchausleihe enthält (über 1000 Entleiher, 200 Bücher mit mehreren tausend Exemplaren und viele Ausleihvorgänge). Die Datenbank kann in OpenOffice eingebunden werden.</p>
<p><b>2. Modellierung von relationalen Datenbanken</b></p> <p>(a) Entity-Relationship-Diagramm</p> <ul style="list-style-type: none"> <li>• Ermittlung von Entitäten, zugehörigen Attributen, Relationen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms</li> <li>• Erläuterung und Modifizierung einer Datenbankmodellierung</li> </ul> <p>(b) Entwicklung einer Datenbank aus einem Datenbankentwurf</p> <ul style="list-style-type: none"> <li>• Modellierung eines relationalen Datenbankschemas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln</li> </ul> <p>(c) Normalform</p> <ul style="list-style-type: none"> <li>• Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten)</li> </ul>		
<p><b>3. Implementierung einer relationalen Datenbank (nur LK)</b></p>		

## Unterrichtsvorhaben Q1-V:

**Thema:** Sicherheit und Datenschutz in Netzstrukturen, Client – Server Kommunikation

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

### Vorhabenbezogene Konkretisierung:

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

**Zeitbedarf:** 10 Stunden(GK) / 25 Stunden (LK)

### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<b>1. Daten in Netzwerken und Sicherheitsaspekte in Netzen sowie beim Zugriff auf Datenbanken</b> (a) Beschreibung eines Datenbankzugriffs im Netz anhand eines Anwendungskontextes und einer Client-Server-Struktur zur Klärung der Funktionsweise eines Datenbankzugriffs (b) Netztopologien als Grundlage von Client-Server-Strukturen und TCP/IP-Schichtenmodell als Beispiel für eine Paketübermittlung in einem Netz (c) Vertraulichkeit, Integrität, Authentizität in Netzwerken sowie symmetrische und asymmetrische kryptografische Verfahren (Cäsar-, Vigenère-, RSA-Verfahren) als Methoden Daten im Netz verschlüsselt zu übertragen	Die Schülerinnen und Schüler <ul style="list-style-type: none"><li>• beschreiben und erläutern Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A),</li><li>• analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A),</li><li>• 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),</li><li>• 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),</li><li>• nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D).</li></ul>	<i>Filius</i>  <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben, Verschlüsselung Q1.5 - Zugriff auf Daten in Netzwerken  <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1 5 - Datenschutz beim Videocenter, Materialblatt-Datenschutzgesetz
<b>2. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht</b>		
<b>3. Implementierung eines Client-Server-Projekts (nur LK)</b>		

## Unterrichtsvorhaben Q2-I:

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

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

### Vorhabenbezogene Konkretisierung:

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

Unter anderem sollen die verschiedenen Baumtraversierungen (Pre-, Post- und Inorder) implementiert werden. Unterschiede bezüglich der Möglichkeit, den Baum anhand der Ausgabe der Baumhalte via Pre-, In- oder Postorder-Traversierung zu rekonstruieren, werden dabei ebenfalls angesprochen, indem die fehlende Umkehrbarkeit der Zuordnung Binärbaum → Inorder-Ausgabe an einem Beispiel verdeutlicht wird.

Eine Tiefensuche wird verwendet, um einen in der Baumstruktur gespeicherten Inhalt zu suchen.

Zu einer Problemstellung in einem entsprechenden Anwendungskontext werden die Operationen der Datenstruktur Suchbaum thematisiert und unter der Verwendung der Klasse `BinarySearchTree` (der Materialien für das Zentralabitur in NRW) weitere Klassen oder Methoden in diesem Anwendungskontext modelliert und implementiert. Auch in diesem Kontext werden grafische Darstellungen der Bäume verwendet.

Die Verwendung von binären Bäumen und Suchbäumen wird anhand weiterer Problemstellungen oder anderen Kontexten weiter geübt.

**Zeitbedarf:** 24 Stunden(GK) / 35 Stunden (LK)

### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<b>1. Analyse von Baumstrukturen in verschiedenen Kontexten</b> (a) Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit) (b) Aufbau und Darstellung von binären Bäumen anhand	Die Schülerinnen und Schüler <ul style="list-style-type: none"><li>• erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A),</li><li>• analysieren und erläutern Algorithmen und</li></ul>	<i>Beispiel:</i> Ternbaum Der Aufbau von Termen wird mit Hilfe von binären Baumstrukturen verdeutlicht. oder

<p>von Baumstrukturen in verschiedenen Kontexten</p>	<p>Programme (A),</p> <ul style="list-style-type: none"> <li>• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li> <li>• ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M),</li> <li>• modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),</li> <li>• verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M),</li> <li>• entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M),</li> <li>• implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),</li> <li>• modifizieren Algorithmen und Programme (I),</li> <li>• nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>• testen Programme systematisch anhand von Beispielen (I),</li> <li>• stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D),</li> <li>• stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).</li> </ul>	<p><i>Beispiel:</i> Ahnenbaum</p> <p><i>Beispiel:</i> Suchbäume (zur sortierten Speicherung von Daten)</p> <p><i>Beispiel:</i> Entscheidungsbäume</p> <p><i>oder</i></p> <p><i>Beispiel:</i> Codierungsbäume</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Binärbaum</p> <p><i>Beispiel:</i> Informatikerbaum <i>als binärer Baum</i> 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</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Binärbaum</p> <p><i>Beispiel:</i> Informatikerbaum <i>als Suchbaum</i></p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Binärer Suchbaum</p>
<p><b>2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse <code>BinaryTree</code></b></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>(e) Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf</p>		
<p><b>3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse <code>BinarySearchTree</code></b></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramm, grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften</p> <p>(c) Erarbeitung der Klasse <code>BinarySearchTree</code> und Einführung des Interface <code>Item</code> zur Realisierung einer geeigneten Ordnungsrelation</p> <p>(d) Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums</p>		
<p><b>4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen</b></p>		

---

## Unterrichtsvorhaben Q2-II:

**Thema:** Endliche Automaten und formale Sprachen

**Leitfragen:** *Wie kann man (endliche) Automaten genau beschreiben? Wie können endliche Automaten (in alltäglichen Kontexten oder zu informatischen Problemstellungen) modelliert werden? Wie können Sprachen durch Grammatiken beschrieben werden? Welche Zusammenhänge gibt es zwischen formalen Sprachen, endlichen Automaten und regulären Grammatiken?*

### **Vorhabenbezogene Konkretisierung:**

Anhand kontextbezogener Beispiele werden endliche Automaten entwickelt, untersucht und modifiziert. Dabei werden verschiedene Darstellungsformen für endliche Automaten ineinander überführt und die akzeptierten Sprachen endlicher Automaten ermittelt. An einem Beispiel wird ein nichtdeterministischer Akzeptor eingeführt als Alternative gegenüber einem entsprechenden deterministischen Akzeptor.

Anhand kontextbezogener Beispiele werden Grammatiken regulärer Sprachen entwickelt, untersucht und modifiziert. Der Zusammenhang zwischen regulären Grammatiken und endlichen Automaten wird verdeutlicht durch die Entwicklung von allgemeinen Verfahren zur Erstellung einer regulären Grammatik für die Sprache eines gegebenen endlichen Automaten bzw. zur Entwicklung eines endlichen Automaten, der genau die Sprache einer gegebenen regulären Grammatik akzeptiert.

Auch andere Grammatiken werden untersucht, entwickelt oder modifiziert. An einem Beispiel werden die Grenzen endlicher Automaten ausgelotet.

**Zeitbedarf:** 20 Stunden (GK) / 40 Stunden (LK)

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p><b>1. Endliche Automaten</b></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"> <li>• analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens auf bestimmte Eingaben (A),</li> <li>• analysieren und erläutern Grammatiken regulärer Sprachen (A),</li> </ul>	<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><b>2. Untersuchung und Entwicklung von Grammatiken regulärer Sprachen</b></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"> <li>• zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A),</li> <li>• ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A),</li> <li>• entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M),</li> <li>• entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M),</li> <li>• entwickeln zur akzeptierten Sprache eines Automaten die zugehörige Grammatik (M),</li> <li>• entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M),</li> <li>• modifizieren Grammatiken regulärer Sprachen (M),</li> <li>• entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M),</li> </ul>	<p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.2 – Endliche Automaten, Formale Sprachen</p> <p><i>Beispiele:</i> reguläre Grammatik für Wörter mit ungerader Parität, Grammatik für Wörter, die bestimmte Zahlen repräsentieren, Satzgliedergrammatik</p>
<p><b>3. Grenzen endlicher Automaten</b></p>	<ul style="list-style-type: none"> <li>• stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D),</li> </ul>	<p><i>Beispiele:</i> Klammerausdrücke, <math>a^n b^n</math> im Vergleich zu <math>(ab)^n</math></p>
<p><b>4. Kellerautomaten und kontextfreie Sprachen</b></p>	<ul style="list-style-type: none"> <li>• ermitteln die Sprache, die ein endlicher Automat akzeptiert (D).</li> </ul>	
<p><b>5. Modellierung und Implementierung eines Scanners, Parsers und Interpreters (nur LK)</b></p>	<ul style="list-style-type: none"> <li>• beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D).</li> </ul>	

## Unterrichtsvorhaben Q2-III:

**Thema:** Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

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

### Vorhabenbezogene Konkretisierung:

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

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

**Zeitbedarf:** 12 Stunden(GK) / 15 Stunden (LK)

### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<b>1. Von-Neumann-Architektur und die Ausführung maschinennaher Programme</b> a) prinzipieller Aufbau einer von Neumann-Architektur mit CPU, Rechenwerk, Steuerwerk, Register und Hauptspeicher b) einige maschinennahe Befehle und ihre Repräsentation in einem Binär-Code, der in einem Register gespeichert werden kann c) Analyse und Erläuterung der Funktionsweise eines einfachen maschinennahen Programms	Die Schülerinnen und Schüler <ul style="list-style-type: none"><li>• erläutern die Ausführung eines einfachen maschinennahen Programms sowie die Datenspeicherung auf einer „Von-Neumann-Architektur“ (A),</li><li>• untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A).</li></ul>	<i>Beispiel:</i> Addition von 4 zu einer eingegebenen Zahl mit einem Rechnermodell  <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 –Von-Neumann-Architektur und maschinennahe Programmierung
<b>2. Grenzen der Automatisierbarkeit</b> a) Vorstellung des Halteproblems b) Unlösbarkeit des Halteproblems c) Beurteilung des Einsatzes von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen		<i>Beispiel:</i> Halteproblem  <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 - Halteproblem
<b>3. Benutzeroberflächen zur Kommunikation in einem Informatiksystem! (nur LK)</b>		

---

**Unterrichtsvorhaben Q2-IV:**

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