

SCHULINTERNER LEHRPLAN  
ZUM KERNLEHRPLAN FÜR DIE GYMNASIALE OBERSTUFE

# Informatik

Europaschule Ostendorf-Gymnasium

Zuletzt aktualisiert am:  
31. Mai 2022

## Inhaltsverzeichnis

1	Die Fachgruppe Informatik der Europaschule Ostendorf-Gymnasium.....	3
2	Unterrichtsvorhaben .....	5
2.1	Übersichtsraster Unterrichtsvorhaben.....	5
2.1.1	Einführungsphase.....	5
2.1.2	Qualifikationsphase – Grundkurs.....	7
2.1.3	Qualifikationsphase – Leistungskurs.....	9
2.2	Konkretisierungsebene.....	11
2.2.1	Einführungsphase.....	11
2.2.2	Qualifikationsphase 1 - Grundkurs.....	21
2.2.3	Qualifikationsphase 2 – Grundkurs .....	30
2.2.4	Qualifikationsphase 1 - Leistungskurs.....	38
2.2.5	Qualifikationsphase 2 - Leistungskurs.....	47
2.3	Grundsätze der fachmethodischen und fachdidaktischen Arbeit .....	54
2.4	Grundsätze der Leistungsbewertung und Leistungsrückmeldung .....	54

## 1 Die Fachgruppe Informatik der Europaschule Ostendorf-Gymnasium

Trotz Innenstadtlage ist das direkte Schulumfeld durch die Insellage zwischen Lippe und Lippe-Kanal ruhig und naturnah. Die Kernstadt Lippstadt ist städtisch-industriell geprägt, die zugehörigen Ortsteile eher dörflich-ländlich. Ein größeres Erholungsgebiet und die typischen städtischen Einrichtungen sind mit dem Rad oder mit öffentlichen Verkehrsmitteln und z.T. zu Fuß leicht erreichbar. Das Kulturprogramm der Stadt bietet ein attraktives kinder- und jugendkulturelles Angebot von theaterpädagogisch begleiteten Vorstellungen des städtischen Jugendtheaters über Lesungen zeitgenössischer Kinder- und Jugendbuchautor\*innen bis hin zu Poetry Slams und Camps zur Förderung der Medienkompetenz von Kindern und Jugendlichen. Das benachbarte Stadttheater kann die Schule für Veranstaltungen, Aufführungen und Konzerte nutzen.

Die Hauptstelle der städtischen Bibliothek bietet die übliche Buch- und Medienausleihe.

Im Bereich der Leistungskurse der Sekundarstufe II kooperiert die Europaschule Ostendorf-Gymnasium mit seinen Nachbarschulen und bietet mit ihnen zahlreiche gemeinsame Kurse an, unter anderem auch Informatik.

Das Fach Informatik wird ab dem Schuljahr 2022/23 im Jahrgang 6 zweistündig unterrichtet. Außerdem haben die Schüler:innen die Möglichkeit, im Wahlpflichtbereich II (WP II) das Fach Informatik zu wählen, welches dann dreistündig unterrichtet wird.

Es wird versucht, die Inhalte der Oberstufen-Informatik und des WP II-Kurses Informatik weitestgehend zu trennen. Dadurch ist die Wahl des Faches Informatik in der Sekundarstufe II unabhängig davon, ob in der Mittelstufe bereits der entsprechende WP II-Kurs gewählt wurde.

In unserem Schulprogramm ist als wesentliches Ziel der Schule beschrieben, die Lernenden als Individuen mit jeweils besonderen Fähigkeiten, Stärken und Interessen in den Blick zu nehmen. Es ist ein wichtiges Anliegen, durch gezielte Unterstützung des Lernens die Potenziale jeder Schülerin und jedes Schülers in allen Bereichen optimal zu entwickeln. Bei der Arbeit mit Informatiksystemen erhalten die Lernenden regelmäßige Rückmeldungen über die Korrektheit ihrer Lösungen und damit auch über ihren individuellen Lernfortschritt. Durch Öffnung von Aufgabenstellungen oder Anregungen der Lehrperson können individuelle Interessen berücksichtigt und weitergehende Kompetenzen erworben werden. In einem längerfristigen Entwicklungsprozess arbeitet das Fach Informatik daran, die Bedingungen für erfolgreiches und individuelles Lernen zu verbessern.

Das Fach Informatik ermöglicht Einsicht in den Aufbau, die Funktion und Nutzung von Informatiksystemen und leistet damit einen wesentlichen Beitrag zur Bildung in der digitalen Welt, der auch einen wesentlichen Punkt des Schulprogrammes darstellt. Die Lernenden werden damit zu einem kompetenten und reflektierten Umgang mit Informatiksystemen befähigt.

### **Verfügbare Ressourcen**

Die Schule verfügt über ein stabiles WLAN-Netz im gesamten Gebäude. In den Unterrichtsräumen stehen generell Dokumentenkamera, Beamer und Apple-TV sowie Lautsprecher zur Nutzung bereit. Allen Lehrkräften steht ein Dienst-iPad zur Verfügung, mobile iPad-Koffer sind zur stundenweisen

Nutzung in Lerngruppen entleihbar. Es stehen darüber hinaus noch ein Computerraum sowie 30 iPads mit Tastatur für den Informatikunterricht zur Verfügung.

Ab der Klasse 7 verfügen die Schüler:innen über in der Regel elternfinanzierte eigene iPads, die zentral administriert werden. Auf den Geräten sind jeweils die für den jeweiligen Jahrgang benötigten fachspezifischen Apps installiert, als digitales Schulheft wird GoodNotes verwendet.

Alle Schüler:innen und Lehrkräfte sind in die schulische Lernplattform IServ eingebunden und verfügen über schulische E-Mail-Adressen. Über IServ ist zudem der Stunden- und Vertretungsplan einsehbar. Über das Aufgabentool laden Schüler:innen bearbeitete Aufgaben hoch und erhalten per Text- oder Sprachdatei Rückmeldung durch die Lehrkraft oder Einsicht in Musterlösungen. Die Klassen und Kurse können über einen Messenger innerhalb der Lerngruppe kommunizieren und auch kollaborativ an Dokumenten arbeiten. Die Lernplattform bietet Textverarbeitung und Tabellenkalkulation sowie die Möglichkeit zur Videokonferenz.

Ansprechpartnerin für die schulische Medienausstattung: Antje Bornhöft

Zurzeit besteht die Fachschaft Informatik aus drei Lehrkräften, denen ein Computerraum mit 16 Computerarbeitsplätzen und ein weiterer Raum mit 30 Tablets zur Verfügung stehen.

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

#### **Funktionsinhaber/innen der Fachgruppe**

*Fachkonferenzvorsitzender:*

Thomas Arbeiter

*Stellvertreterin:*

Monika Pott

*Kontaktlehrer für die Zusammenarbeit mit außerschulischen Partnern:*

Thomas Arbeiter

*Ansprechpartner für Informatik-Biber-Wettbewerb:*

Thomas Arbeiter

*Ansprechpartnerin für Jugendwettbewerb Informatik (JwInf):*

Monika Pott

*Ansprechpartner für Bundeswettbewerb Informatik (BwInf):*

Thomas Arbeiter

*Ansprechpartner für sonstige Wettbewerbe:*

Monika Pott /  
Thomas Arbeiter

## 2 Unterrichtsvorhaben

### 2.1 Übersichtsraster Unterrichtsvorhaben

#### 2.1.1 Einführungsphase

	<b>Inhaltlicher Schwerpunkt</b>	<b>Inhaltsfelder</b>	<b>Zentrale Kompetenzen</b>
Unterrichtsvorhaben EF-I: <b>Was macht Informatik? – Einführung in die Inhaltsfelder der Informatik</b>	<ul style="list-style-type: none"> <li>- Einsatz, Nutzung und Aufbau von Informatiksystemen</li> <li>- Wirkung der Automatisierung</li> </ul>	<ul style="list-style-type: none"> <li>- Informatiksysteme</li> <li>- Informatik, Mensch und Gesellschaft</li> </ul>	<ul style="list-style-type: none"> <li>- Kommunizieren und Kooperieren</li> <li>- Darstellen und Interpretieren</li> <li>- Argumentieren</li> </ul>
Unterrichtsvorhaben EF-II: <b>Grundlagen der objektorientierten Analyse, Modellierung und Implementierung</b>	<ul style="list-style-type: none"> <li>- Objekte und Klassen</li> <li>- Syntax und Semantik einer Programmiersprache</li> </ul>	<ul style="list-style-type: none"> <li>- Daten und ihre Strukturierung</li> <li>- Formale Sprachen und Automaten</li> </ul>	<ul style="list-style-type: none"> <li>- Modellieren</li> <li>- Implementieren</li> <li>- Darstellen und Interpretieren</li> <li>- Kommunizieren und Kooperieren</li> </ul>
Unterrichtsvorhaben EF-III: <b>Algorithmische Grundstrukturen</b>	<ul style="list-style-type: none"> <li>- Objekte und Klassen</li> <li>- Syntax und Semantik einer Programmiersprache</li> <li>- Analyse, Entwurf und Implementierung einfacher Algorithmen</li> </ul>	<ul style="list-style-type: none"> <li>- Daten und ihre Strukturierung</li> <li>- Algorithmen</li> <li>- Formale Sprachen und Automaten</li> </ul>	<ul style="list-style-type: none"> <li>- Argumentieren</li> <li>- Modellieren</li> <li>- Implementieren</li> <li>- Kommunizieren und Kooperieren</li> </ul>
Unterrichtsvorhaben EF-IV: <b>Das ist die digitale Welt! – Einführung in die Grundlagen, Anwendungsgebiete und Verarbeitung binärer Codierung</b>	<ul style="list-style-type: none"> <li>- Binäre Codierung und Verarbeitung</li> <li>- Besondere Eigenschaften der digitalen Speicherung und Verarbeitung von Daten</li> </ul>	<ul style="list-style-type: none"> <li>- Informatiksysteme</li> <li>- Informatik, Mensch und Gesellschaft</li> </ul>	<ul style="list-style-type: none"> <li>- Kommunizieren und Kooperieren</li> <li>- Darstellen und Interpretieren</li> <li>- Argumentieren</li> </ul>
Unterrichtsvorhaben EF-V: <b>Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand lebensnaher Anforderungsbeispiele</b>	<ul style="list-style-type: none"> <li>- Objekte und Klassen</li> <li>- Syntax und Semantik einer Programmiersprache</li> </ul>	<ul style="list-style-type: none"> <li>- Daten und ihre Strukturierung</li> <li>- Algorithmen</li> <li>- Formale Sprachen und Automaten</li> </ul>	<ul style="list-style-type: none"> <li>- Kommunizieren und Kooperieren</li> <li>- Darstellen und Interpretieren</li> <li>- Argumentieren</li> <li>- Modellieren</li> </ul>

	<b>Inhaltlicher Schwerpunkt</b>	<b>Inhaltsfelder</b>	<b>Zentrale Kompetenzen</b>
	- Analyse, Entwurf und Implementierung einfacher Algorithmen		- Implementieren
Unterrichtsvorhaben EF-VI: <b>Such- und Sortieralgorithmen</b>	- Algorithmen zum Suchen und Sortieren - Analyse, Entwurf und Implementierung einfacher Algorithmen - Objekte und Klassen	- Algorithmen - Daten und ihre Strukturierung	- Argumentieren - Modellieren - Darstellen und interpretieren - Kommunizieren und Kooperieren
Unterrichtsvorhaben EF-VII: <b>Leben in der digitalen Welt – Immer mehr Möglichkeiten und immer mehr Gefahren!?</b>	- Geschichte der automatischen Datenverarbeitung - Wirkungen der Automatisierung - Dateisystem	- Informatiksysteme - Informatik und Gesellschaft	- Kommunizieren und Kooperieren - Darstellen und interpretieren - Argumentieren

## 2.1.2 Qualifikationsphase – Grundkurs

	Inhaltlicher Schwerpunkt	Inhaltsfelder	Zentrale Kompetenzen
<b>Qualifikationsphase 1</b>			
Unterrichtsvorhaben GK-Q1-I: <b>Wiederholung und Vertiefung der objektorientierten Modellierung</b>	<ul style="list-style-type: none"> <li>- Objekte und Klassen</li> <li>- Wirkung der Automatisierung</li> <li>- Nutzung von Informatiksystemen</li> </ul>	<ul style="list-style-type: none"> <li>- Daten und ihre Strukturierung</li> <li>- Algorithmen</li> <li>- Informatik, Mensch und Gesellschaft</li> </ul>	<ul style="list-style-type: none"> <li>- Modellieren</li> <li>- Darstellen und Interpretieren</li> <li>- Implementieren</li> <li>- Kommunizieren und Kooperieren</li> </ul>
Unterrichtsvorhaben GK-Q1-II: <b>Organisation und Verarbeitung von Daten I – Modellierung und Implementierung von Anwendungen mit dynamischen linearen Datenstrukturen</b>	<ul style="list-style-type: none"> <li>- Objekte und Klassen</li> <li>- Syntax und Semantik einer Programmiersprache</li> <li>- Analyse, Entwurf und Implementierung von Algorithmen</li> <li>- Algorithmen in ausgewählten informatischen Kontexten</li> </ul>	<ul style="list-style-type: none"> <li>- Daten und ihre Strukturierung</li> <li>- Algorithmen</li> </ul>	<ul style="list-style-type: none"> <li>- Modellieren</li> <li>- Implementieren</li> <li>- Darstellen und Interpretieren</li> <li>- Argumentieren</li> </ul>
Unterrichtsvorhaben GK-Q1-III: <b>Automaten und formale Sprachen</b>	<ul style="list-style-type: none"> <li>- Syntax und Semantik einer Programmiersprache</li> <li>- Endliche Automaten</li> <li>- Grammatiken regulärer Sprachen</li> <li>- Möglichkeiten und Grenzen von Automaten und formalen Sprachen</li> </ul>	<ul style="list-style-type: none"> <li>- Formale Sprachen und Automaten</li> <li>- Informatiksysteme</li> </ul>	<ul style="list-style-type: none"> <li>- Argumentieren</li> <li>- Darstellen und Interpretieren</li> <li>- Modellieren</li> <li>- Kommunizieren und Kooperieren</li> </ul>
Unterrichtsvorhaben GK-Q1-IV: <b>Organisation und Verarbeitung von Daten II – Modellierung und Implementierung von Anwendungen mit dynamischen nicht-linearen Datenstrukturen</b>	<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>	<ul style="list-style-type: none"> <li>- Daten und ihre Strukturierung</li> <li>- Algorithmen</li> <li>- Formale Sprachen und Automaten</li> </ul>	<ul style="list-style-type: none"> <li>- Argumentieren</li> <li>- Darstellen und Interpretieren</li> <li>- Modellieren</li> <li>- Implementieren</li> <li>- Kommunizieren und Kooperieren</li> </ul>

**Qualifikationsphase 2**

<p>Unterrichtsvorhaben GK-Q2-I: <b>Aufbau von und Kommunikation in Netzwerken</b></p>	<ul style="list-style-type: none"> <li>- Einzelrechner und Rechnernetzwerke</li> <li>- Sicherheit</li> <li>- Nutzung von Informatiksystemen, Wirkungen der Automatisierung</li> </ul>	<ul style="list-style-type: none"> <li>- Informatiksysteme</li> <li>- Informatik, Mensch und Gesellschaft</li> </ul>	<ul style="list-style-type: none"> <li>- Argumentieren</li> <li>- Darstellen und Interpretieren</li> <li>- Kommunizieren und Kooperieren</li> </ul>
<p>Unterrichtsvorhaben GK-Q2-II: <b>Nutzung und Modellierung von relationalen Datenbanken in Anwendungskontexten</b></p>	<ul style="list-style-type: none"> <li>- Datenbanken</li> <li>- Algorithmen in ausgewählten informatischen Kontexten</li> <li>- Syntax und Semantik einer Programmiersprache</li> <li>- Sicherheit</li> </ul>	<ul style="list-style-type: none"> <li>- Daten und Strukturierung</li> <li>- Algorithmen</li> <li>- Formale Sprachen und Automaten</li> <li>- Informatik, Mensch und Gesellschaft</li> </ul>	<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>Unterrichtsvorhaben GK-Q2-III: <b>Algorithmen zum Suchen und Sortieren auf linearen Datenstrukturen</b></p>	<ul style="list-style-type: none"> <li>- Analyse, Entwurf und Implementierung von Algorithmen</li> <li>- Algorithmen in ausgewählten informatischen Kontexten</li> <li>- Syntax und Semantik einer Programmiersprache</li> </ul>	<ul style="list-style-type: none"> <li>- Algorithmen</li> <li>- Formale Sprachen und Automaten</li> </ul>	<ul style="list-style-type: none"> <li>- Argumentieren</li> <li>- Darstellen und Interpretieren</li> <li>- Modellieren</li> <li>- Implementieren</li> </ul>
<p>Unterrichtsvorhaben GK-Q2-IV: <b>Informatik und Gesellschaft</b></p>	<ul style="list-style-type: none"> <li>- Datenschutz</li> <li>- Verantwortung in der Informatik</li> </ul>	<ul style="list-style-type: none"> <li>- Informatik, Mensch und Gesellschaft</li> </ul>	<ul style="list-style-type: none"> <li>- Argumentieren</li> <li>- Kommunizieren und Kooperieren</li> </ul>



### 2.1.3 Qualifikationsphase – Leistungskurs

	Inhaltlicher Schwerpunkt	Inhaltsfelder	Zentrale Kompetenzen
<b>Qualifikationsphase 1</b>			
Unterrichtsvorhaben LK-Q1-I: <b>Wiederholung und Vertiefung der objektorientierten Modellierung</b>	<ul style="list-style-type: none"> <li>- Objekte und Klassen</li> <li>- Wirkung der Automatisierung</li> <li>- Nutzung von Informatiksystemen</li> </ul>	<ul style="list-style-type: none"> <li>- Daten und ihre Strukturierung</li> <li>- Algorithmen</li> <li>- Informatik, Mensch und Gesellschaft</li> </ul>	<ul style="list-style-type: none"> <li>- Modellieren</li> <li>- Darstellen und Interpretieren</li> <li>- Implementieren</li> <li>- Kommunizieren und Kooperieren</li> </ul>
Unterrichtsvorhaben LK-Q1-II: <b>Organisation und Verarbeitung von Daten I – Modellierung und Implementierung von Anwendungen mit dynamischen linearen Datenstrukturen</b>	<ul style="list-style-type: none"> <li>- Objekte und Klassen</li> <li>- Syntax und Semantik einer Programmiersprache</li> <li>- Analyse, Entwurf und Implementierung von Algorithmen</li> <li>- Algorithmen in ausgewählten informatischen Kontexten</li> <li>- Wirkungen der Automatisierung</li> </ul>	<ul style="list-style-type: none"> <li>- Daten und ihre Strukturierung</li> <li>- Algorithmen</li> <li>- Informatik, Mensch und Gesellschaft</li> </ul>	<ul style="list-style-type: none"> <li>- Modellieren</li> <li>- Implementieren</li> <li>- Darstellen und Interpretieren</li> <li>- Argumentieren</li> </ul>
Unterrichtsvorhaben LK-Q1-III: <b>Automaten und formale Sprachen</b>	<ul style="list-style-type: none"> <li>- Syntax und Semantik einer Programmiersprache</li> <li>- Endliche Automaten und Kellerautomaten</li> <li>- Grammatiken regulärer und kontextfreier Sprachen</li> <li>- Scanner, Parser und Interpreter für eine reguläre Sprache</li> <li>- Möglichkeiten und Grenzen von Automaten und formalen Sprachen</li> </ul>	<ul style="list-style-type: none"> <li>- Formale Sprachen und Automaten</li> <li>- Informatiksysteme</li> <li>- Informatik, Mensch und Gesellschaft</li> </ul>	<ul style="list-style-type: none"> <li>- Argumentieren</li> <li>- Darstellen und Interpretieren</li> <li>- Modellieren</li> <li>- Implementieren</li> <li>- Kommunizieren und Kooperieren</li> </ul>
Unterrichtsvorhaben LK-Q1-IV: <b>Organisation und Verarbeitung von Daten II – Modellierung und Implementierung von Anwendungen mit dynamischen nicht-linearen Datenstrukturen</b>	<ul style="list-style-type: none"> <li>- Objekte und Klassen</li> <li>- Analyse, Entwurf und Implementierung von Algorithmen</li> </ul>	<ul style="list-style-type: none"> <li>- Daten und ihre Strukturierung</li> <li>- Algorithmen</li> <li>- Formale Sprachen und Automaten</li> </ul>	<ul style="list-style-type: none"> <li>- Argumentieren</li> <li>- Darstellen und Interpretieren</li> <li>- Modellieren</li> <li>- Implementieren</li> </ul>

	Inhaltlicher Schwerpunkt	Inhaltsfelder	Zentrale Kompetenzen
	<ul style="list-style-type: none"> <li>- Algorithmen in ausgewählten informatischen Kontexten</li> <li>- Syntax und Semantik einer Programmiersprache</li> </ul>		<ul style="list-style-type: none"> <li>- Kommunizieren und Kooperieren</li> </ul>
<b>Qualifikationsphase 2</b>			
Unterrichtsvorhaben LK-Q2-I: <b>Aufbau von und Kommunikation in Netzwerken</b>	<ul style="list-style-type: none"> <li>- Einzelrechner und Rechnernetzwerke</li> <li>- Sicherheit</li> <li>- Nutzung von Informatiksystemen</li> <li>- Analyse, Entwurf und Implementierung von Algorithmen</li> <li>- Algorithmen in ausgewählten informatischen Kontexten</li> </ul>	<ul style="list-style-type: none"> <li>- Informatiksysteme</li> <li>- Algorithmen</li> </ul>	<ul style="list-style-type: none"> <li>- Argumentieren</li> <li>- Darstellen und Interpretieren</li> <li>- Modellieren</li> <li>- Implementieren</li> <li>- Kommunizieren und Kooperieren</li> </ul>
Unterrichtsvorhaben LK-Q2-II: <b>Nutzung, Modellierung und Implementation von relationalen Datenbanken in Anwendungskontexten</b>	<ul style="list-style-type: none"> <li>- Datenbanken</li> <li>- Algorithmen in ausgewählten informatischen Kontexten</li> <li>- Syntax und Semantik einer Programmiersprache</li> <li>- Sicherheit</li> </ul>	<ul style="list-style-type: none"> <li>- Daten und Strukturierung</li> <li>- Algorithmen</li> <li>- Formale Sprachen und Automaten</li> <li>- Informatik, Mensch und Gesellschaft</li> </ul>	<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>
Unterrichtsvorhaben LK-Q2-III: <b>Algorithmen zum Suchen und Sortieren auf linearen Datenstrukturen</b>	<ul style="list-style-type: none"> <li>- Analyse, Entwurf und Implementierung von Algorithmen</li> <li>- Algorithmen in ausgewählten informatischen Kontexten</li> <li>- Syntax und Semantik einer Programmiersprache</li> <li>- Wirkung der Automatisierung</li> </ul>	<ul style="list-style-type: none"> <li>- Algorithmen</li> <li>- Formale Sprachen und Automaten</li> <li>- Informatik, Mensch und Gesellschaft</li> </ul>	<ul style="list-style-type: none"> <li>- Argumentieren</li> <li>- Darstellen und Interpretieren</li> <li>- Modellieren</li> <li>- Implementieren</li> </ul>
Unterrichtsvorhaben LK-Q2-IV: <b>Informatik und Gesellschaft</b>	<ul style="list-style-type: none"> <li>- Datenschutz</li> <li>- Verantwortung in der Informatik</li> </ul>	<ul style="list-style-type: none"> <li>- Informatik, Mensch und Gesellschaft</li> </ul>	<ul style="list-style-type: none"> <li>- Argumentieren</li> <li>- Kommunizieren und Kooperieren</li> </ul>

## 2.2 Konkretisierungsebene

### 2.2.1 Einführungsphase

#### 2.2.1.1 Unterrichtsvorhaben EF-I

**Thema:** Was macht Informatik? – Einführung in die Inhaltsfelder der Informatik

**Leitfragen:** Was macht Informatik? Welche fundamentalen Konzepte müssen Informatiker:innen in ihre Arbeit einbeziehen, damit informatische Systeme effizient und zuverlässig arbeiten können? Wo lassen sich diese Konzepte (in Ansätzen) in dem schuleigenen Netzwerk und Computersystem wiederfinden?

#### **Vorhabenbezogene Konkretisierung:**

Im ersten Unterrichtsvorhaben werden die fünf Inhaltsfelder des Faches Informatik beispielhaft an einem Informatiksystem erarbeitet. Das Unterrichtsvorhaben ist so strukturiert, dass die Schülerinnen und Schüler anhand bekannter Alltagstechnik die Grundideen fundamentaler informatischer Konzepte (Inhaltsfelder) größtenteils selbstständig erarbeiten und nachvollziehen.

Ausgehend von dem bekannten Bedienungs- und Funktionalitätswissen eines Navigationsgerätes werden die Strukturierung von Daten, das Prinzip der Algorithmik, die Eigenheit formaler Sprachen, die Kommunikationsfähigkeit von Informatiksystemen und die positiven und negativen Auswirkungen auf Mensch und Gesellschaft thematisiert. Das am Beispiel Navigationsgerät erworbene Wissen kann auf weitere den Schülerinnen und Schülern bekannte Informatiksysteme übertragen werden.

In einem letzten Schritt kann ausgehend von den Inhaltsfeldern das Schulnetzwerk in Ansätzen so analysiert werden, dass ein kompetenter Umgang mit diesem ermöglicht wird.

**Zeitbedarf:** 6 Stunden

#### **Sequenzierung des Unterrichtsvorhabens:**

<b>Unterrichtssequenzen</b>	<b>Zu entwickelnde Kompetenzen</b>	<b>Kapitel und Materialien</b>
<b>1. Informatiksysteme und ihr genereller Aufbau</b> (a) Daten und ihre Strukturierung (b) Algorithmen (c) Formale Sprachen und Automaten (d) Informatiksysteme (e) Informatik, Mensch und Gesellschaft	Die Schülerinnen und Schüler - bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A) - nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D)	<b>Kapitel 1 Was macht Informatik?</b> Als Anschauungsmaterial bieten sich Navigationsgeräte und Landkarten an.
<b>2. Der kompetente Umgang mit dem Schulnetzwerk</b> (a) Erstellen und Anlegen von Ordnerstrukturen (b) Sortieren von Dateien und Ordnern (c) Eingabe von Befehlen über Eingabeaufforderung (d) Einzelrechner und Netzwerk (e) Sicherheit und Datenschutz		<b>Kapitel 1 Was macht Informatik?</b> Interview mit dem Netzwerkadministrator, Benutzer- und Datenschutzbestimmungen der Schule

### 2.2.1.2 Unterrichtsvorhaben EF-II

**Thema:** Grundlagen der objektorientierten Analyse, Modellierung und Implementierung

**Leitfragen:** Wie lassen sich Gegenstandsbereiche informatisch modellieren und in einem Greenfoot-Szenario informatisch realisieren?

#### **Vorhabenbezogene Konkretisierung:**

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

Dazu werden zunächst konkrete Gegenstandsbereiche aus der Lebenswelt der Schülerinnen und Schüler analysiert und im Sinne des objektorientierten Paradigmas strukturiert. Dabei werden die grundlegenden Begriffe der Objektorientierung und Modellierungswerkzeuge wie Objektdiagramme und Klassendiagramme eingeführt.

Im Anschluss wird die objektorientierte Analyse für das Greenfoot-Szenario Planetenerkundung durchgeführt. Die vom Szenario vorgegebenen Klassen werden von Schülerinnen und Schülern in Teilen analysiert und entsprechende Objekte anhand einfacher Problemstellungen erprobt. Die Lernenden implementieren und testen einfache Programme. Die Greenfoot-Umgebung ermöglicht es, Beziehungen zwischen Klassen zu einem späteren Zeitpunkt (Kapitel 5) zu thematisieren. So kann der Fokus hier auf Grundlagen wie der Unterscheidung zwischen Klasse und Objekt, Attribute, Methoden, Objektidentität und Objektzustand gelegt werden.

Da in Kapitel 2 zudem auf die Verwendung von Kontrollstrukturen verzichtet wird und der Quellcode aus einer rein linearen Sequenz besteht, ist auf diese Weise eine Fokussierung auf die Grundlagen der Objektorientierung möglich, ohne dass algorithmische Probleme ablenken. Natürlich kann die Arbeit an diesen Projekten unmittelbar zum nächsten Unterrichtsvorhaben (Kapitel 3) führen. Dort stehen Kontrollstrukturen im Mittelpunkt.

**Zeitbedarf:** 8 Stunden

#### **Sequenzierung des Unterrichtsvorhabens:**

<b>Unterrichtssequenzen</b>	<b>Zu entwickelnde Kompetenzen</b>	<b>Kapitel und Materialien</b>
<b>1. Identifikation von Objekten und Klassen</b> (a) An einem lebensweltnahen Beispiel werden Objekte und Klassen im Sinne der objektorientierten Modellierung eingeführt. (b) Objekte werden durch Objektdiagramme, Klassen durch Klassendiagramme dargestellt. (c) Die Modellierungen werden einem konkreten Anwendungsfall entsprechend angepasst.	Die Schülerinnen und Schüler - ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften und ihre Operationen (M), - stellen den Zustand eines Objekts dar (D), - modellieren Klassen mit ihren Attributen und ihren Methoden (M), - implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),	<b>Kapitel 2 Einführung in die Objektorientierung</b> 2.1 Objektorientierte Modellierung

<p><b>2. Analyse von Objekten und Klassen im Greenfoot-Szenario</b></p> <p>(a) Schritte der objektorientierten Analyse, Modellierung und Implementation</p> <p>(b) Analyse und Erprobung der Objekte im Greenfoot-Szenario</p>	<p>- implementieren Klassen in einer Programmiersprache, auch unter Nutzung dokumentierter Klassenbibliotheken (I).</p>	<p><b>Kapitel 2 Einführung in die Objektorientierung</b></p> <p>2.2 Das Greenfoot-Szenario „Planetenerkundung“</p> <ul style="list-style-type: none"> <li>• Von der Realität zu Objekten</li> <li>• Von den Objekten zu Klassen</li> <li>• Klassendokumentation</li> <li>• Objekte inspizieren</li> <li>• Methoden aufrufen</li> <li>• Objektidentität und Objektzustand</li> </ul>
<p><b>3. Implementierung einfacher Aktionen in Greenfoot</b></p> <p>(a) Quelltext einer Java-Klasse</p> <p>(b) Implementation eigener Methoden, Dokumentation mit JavaDoc</p> <p>(c) Programme übersetzen (Aufgabe des Compilers) und testen</p>		<p><b>Kapitel 2 Einführung in die Objektorientierung</b></p> <p>2.3 Programmierung in Greenfoot</p> <ul style="list-style-type: none"> <li>• Methoden schreiben</li> <li>• Programme übersetzen und testen</li> </ul>

### 2.2.1.3 Unterrichtsvorhaben EF-III

**Thema:** Algorithmische Grundstrukturen in Java

**Leitfragen:** Wie lassen sich Aktionen von Objekten flexibel realisieren?

**Vorhabenbezogene Konkretisierung:**

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

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

**Zeitbedarf:** 18 Stunden

## Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<b>1. Algorithmen</b> (a) Wiederholungen (While-Schleife, Zählschleife) (b) bedingte Anweisungen (c) Verknüpfung von Bedingungen durch die logischen Funktionen UND, ODER und NICHT (d) Systematisierung des Vorgehens zur Entwicklung von Algorithmen zur Lösung komplexerer Probleme	Die Schülerinnen und Schüler - analysieren und erläutern einfache Algorithmen und Programme (A), - entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M), - ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen zu (M), - modifizieren einfache Algorithmen und Programme (I), - implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I), - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), - implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), - testen Programme schrittweise anhand von Beispielen (I), - interpretieren Fehlermeldungen und korrigieren den Quellcode (I).	<b>Kapitel 3 Algorithmen</b> 3.1 Kontrollstrukturen 3.2 Wiederholungen 3.3 Zählschleifen 3.4 Bedingte Anweisungen 3.5 Logische Operationen 3.6 Algorithmen entwickeln
<b>2. Variablen und Methoden</b> (a) Implementierung eigener Methoden mit lokalen Variablen, auch zur Realisierung einer Zählschleife (b) Implementierung eigener Methoden mit Parameterübergabe und/oder Rückgabewert (c) Implementierung von Konstrukturen (d) Realisierung von Attributen	(siehe oben)	<b>Kapitel 4 Variablen und Methoden</b> 4.1 Variablen 4.2 Methoden

### 2.2.1.4 Unterrichtsvorhaben EF-IV

**Thema:** Das ist die digitale Welt! – Einführung in die Grundlagen, Anwendungsgebiete und Verarbeitung binärer Codierung

**Leitfragen:** Wie werden binäre Informationen gespeichert und wie können sie davon ausgehend weiter verarbeitet werden? Wie unterscheiden sich analoge Medien und Geräte von digitalen Medien und Geräten? Wie ist der Grundaufbau einer digitalen Rechenmaschine?

#### Vorhabenbezogene Konkretisierung:

Das Unterrichtsvorhaben hat die binäre Speicherung und Verarbeitung sowie deren Besonderheiten zum Inhalt.

Im ersten Schritt erarbeiten die Schülerinnen und Schüler anhand ihnen bekannter technischer Gegenstände die Gemeinsamkeiten, Unterschiede und Besonderheiten der jeweiligen analogen und digitalen Version. Nach dieser ersten grundlegenden Einordnung des digitalen Prinzips wenden die Schülerinnen und Schüler das Binäre als Zahlensystem mit arithmetischen und logischen Operationen an und codieren Zeichen binär.

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

**Zeitbedarf:** 8 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

<b>Unterrichtssequenzen</b>	<b>Zu entwickelnde Kompetenzen</b>	<b>Kapitel und Materialien</b>
<b>1. Analoge und digitale Aufbereitung und Verarbeitung von Daten</b> (a) Erarbeitung der Unterschiede von analog und digital (b) Zusammenfassung und Bewertung der technischen Möglichkeiten von analog und digital	Die Schülerinnen und Schüler - bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A) - stellen ganze Zahlen und Zeichen in Binärcodes dar (D) - interpretieren Binärcodes als Zahlen und Zeichen (D) - beschreiben und erläutern den	<b>Die digitale Welt 001 – Von analog zu digital</b>
<b>2. Der Umgang mit binärer Codierung von Informationen</b> (a) Das binäre (und hexadezimale) Zahlensystem (b) Binäre Informationsspeicherung (c) Binäre Verschlüsselung (d) Implementation eines Binärrechners	strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A) - nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K) - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I)	<b>Die digitale Welt 010 – Binäre Welt</b>
<b>3. Aufbau informatischer Systeme</b> (a) Identifikation des EVA-Prinzips als grundlegende Arbeitsweise informatischer Systemen (b) Nachvollziehen der von-Neumann-Architektur als relevantes Modell der Umsetzung des EVA-Prinzips		<b>Die digitale Welt 011 – Der Von-Neumann-Rechner</b>

### 2.2.1.5 Unterrichtsvorhaben EF-V

**Thema:** Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand lebensnaher Anforderungsbeispiele

**Leitfragen:** Wie werden realistische Systeme anforderungsspezifisch reduziert, als Entwurf modelliert und implementiert? Wie kommunizieren Objekte und wie wird dieses dargestellt und realisiert?

### Vorhabenbezogene Konkretisierung:

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

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

Nach diesem ersten Modellierungsschritt werden über Klassendokumentationen und der Darstellung von Objektkommunikationen anhand von Sequenzdiagrammen Implementationsdiagramme entwickelt. Danach werden die Implementationsdiagramme unter Berücksichtigung der Klassendokumentationen in Javaklassen programmiert. In einem letzten Schritt wird das Konzept der Vererbung sowie seiner Vorteile erarbeitet.

Schließlich sind die Schülerinnen und Schüler in der Lage, eigene kleine Softwareprojekte zu entwickeln. Ausgehend von der Dekonstruktion und Erweiterung eines Spiels wird ein weiteres Projekt von Grund auf modelliert und implementiert. Dabei können arbeitsteilige Vorgehensweisen zum Einsatz kommen. In diesem Zusammenhang wird auch das Erstellen von graphischen Benutzeroberflächen eingeführt.

**Zeitbedarf:** 18 Stunden

### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<b>1. Umsetzung von Anforderungen in Entwurfsdiagramme</b> (a) Aus Anforderungsbeschreibungen werden Objekte mit ihren Eigenschaften identifiziert (b) Gleichartige Objekte werden in Klassen (Entwurf) zusammengefasst und um Datentypen und Methoden erweitert	Die Schülerinnen und Schüler - analysieren und erläutern eine objektorientierte Modellierung (A), - stellen die Kommunikation zwischen Objekten 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 in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), - testen Programme schrittweise anhand von Beispielen (I),	<b>Kapitel 5 Klassenentwurf</b> 5.1 Von der Realität zum Programm 5.2 Objekte 5.3 Klassen und Beziehungen entwerfen
<b>2. Implementationsdiagramme als erster Schritt der Programmierung</b> (a) Erweiterung des Entwurfsdiagramms um Konstruktoren und get- und set-Methoden (b) Festlegung von Datentypen in Java, sowie von Rückgaben und Parametern (c) Entwicklung von Klassendokumentationen (d) Erstellung von Sequenzdiagrammen als Vorbereitung für die Programmierung		<b>Kapitel 5 Klassenentwurf</b> 5.4 Klassen und Beziehungen implementieren
<b>3. Programmierung anhand der</b>		<b>Kapitel 5 Klassenentwurf</b>



<p><b>Dokumentation und des Implementations- und Sequenzdiagrammes</b></p> <p>(a) Klassen werden in Java-Quellcode umgesetzt.</p> <p>(b) Das Geheimnisprinzip wird umgesetzt.</p> <p>(c) Einzelne Klassen und das Gesamtsystem werden anhand der Anforderungen und Dokumentationen auf ihre Korrektheit überprüft.</p> <p><b>4. Vererbungsbeziehungen</b></p> <p>(a) Das Grundprinzip der Vererbung wird erarbeitet.</p> <p>(b) Die Vorteile der Vererbungsbeziehungen</p> <p>(c) Vererbung wird implementiert</p>	<ul style="list-style-type: none"> <li>- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>- analysieren und erläutern einfache Algorithmen und Programme (A)</li> <li>- modifizieren einfache Algorithmen und Programme (I),</li> <li>- entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M).</li> <li>- stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D),</li> <li>- dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D)</li> </ul>	<p>5.4 Klassen und Beziehungen implementieren</p> <p><b>Kapitel 5 Klassenentwurf</b> 5.5 Vererbung</p>
<p><b>5. Softwareprojekt</b></p> <p>(a) Analyse und Dekonstruktion eines Spiels (Modelle, Quelltexte)</p> <p>(b) Erweiterung des Spiels um weitere Funktionalitäten</p> <p>(c) Modellierung eines Spiels aufgrund einer Anforderungsbeschreibung, inklusive einer grafischen Benutzeroberfläche</p> <p>(d) (arbeitsteilige) Implementation des Spiels</p>		<p><b>Kapitel 7 Softwareprojekte</b> 7.1 Softwareentwicklung 7.2 Oberflächen</p>

### 2.2.1.6 Unterrichtsvorhaben EF-VI

**Thema:** Such- und Sortieralgorithmen

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

**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 lernen die Schülerinnen und Schüler das Feld als eine erste Datensammlung kennen. Alle Algorithmen dieses Kapitels arbeiten auf einem Feld.

Die Schülerinnen und Schüler lernen zunächst Strategien des Suchens (lineare Suche, binäre Suche, Hashing) und dann des Sortierens (Selection Sort, Insertion Sort, Bubble Sort) kennen.

Daran anschließend lernen die Schülerinnen und Schüler zunächst Strategien des Suchens (lineare Suche, binäre Suche, Hashing) und dann des Sortierens (Selection Sort, Insertion Sort, Bubble Sort)

kennen. Die Projekteinstiege dienen dazu, die jeweiligen Strategien handlungsorientiert zu erkunden und intuitive Effizienzbetrachtungen der Suchalgorithmen vorzunehmen.

Schließlich wird die Effizienz unterschiedlicher Sortierverfahren beurteilt.

**Zeitbedarf:** 9 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Modellierung und Implementation von Datenansammlungen</b></p> <p>(a) Modellierung von Attributen als Felder</p> <p>(b) Deklaration, Instanziierung und Zugriffe auf ein Feld</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>- analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D)</li> <li>- entwerfen einen weiteren Algorithmus zum Sortieren (M)</li> <li>- beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeit und Speicherplatzbedarf (A)</li> </ul>	<p><b>Kapitel 6 Suchen und Sortieren</b></p> <p>6.1 Speichern mit Struktur – Arrays</p>
<p><b>2. Explorative Erarbeitung von Suchverfahren</b></p> <p>(a) Erkundung von Strategien für das Suchen auf unsortierten Daten, auf sortierten Daten und mithilfe einer Berechnungsfunktion</p> <p>(b) Vergleich der drei Verfahren durch intuitive Effizienzbetrachtungen</p>	<ul style="list-style-type: none"> <li>- ordnen Attributen lineare Datenansammlungen zu (M)</li> </ul>	<p><b>Kapitel 6 Suchen und Sortieren</b></p> <p>Projekteinstieg 1: Suchen</p> <p>6.2 Suchen mit System</p> <p>Lineare Suche</p> <p>Binäre Suche</p> <p>Hashing</p>
<p><b>3. Systematisierung von Algorithmen und Effizienzbetrachtungen</b></p> <p>(a) Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode</p> <p>(b) Anwendung von Sortieralgorithmen auf verschiedene Beispiele</p> <p>(c) Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche</p> <p>(d) Analyse eines weiteren Sortieralgorithmus (sofern nicht in (a) bereits geschehen)</p>		<p><b>Kapitel 6 Suchen und Sortieren</b></p> <p>Projekteinstieg 2: Sortieren</p> <p>6.3 Ordnung ist das halbe Leben!? – Sortieren</p> <p>Sortieren</p> <p>Selection Sort</p> <p>Insertion Sort</p> <p>Bubble Sort</p>

### 2.2.1.7 Unterrichtsvorhaben EF-VII

**Thema:** Leben in der digitalen Welt – Immer mehr Möglichkeiten und immer mehr Gefahren!?

**Leitfragen:** Welche Entwicklungen, Ideen und Erfindungen haben zur heutigen Informatik geführt? Welche Auswirkungen hat die Informatik für das Leben des modernen Menschen?

**Vorhabenbezogene Konkretisierung:**

Das Unterrichtsvorhaben stellt die verschiedenen Entwicklungsstränge der Informatik in den Fokus. Darüber hinaus wird beispielhaft analysiert und bewertet, welche Möglichkeiten und Gefahren die moderne Informationsverarbeitung mit sich bringt.

Im ersten Schritt des Unterrichtsvorhabens wird anhand von Themenkomplexen entscheidende Entwicklungen der Informatik erarbeitet. Dabei werden auch übergeordnete Tendenzen identifiziert.

Ausgehend von dieser Betrachtung kann die aktuelle Informatik hinsichtlich ihrer Leistungsfähigkeit analysiert werden. Dabei soll herausgestellt werden, welche positiven und negativen Folgen Informatiksysteme mit sich bringen können.

**Zeitbedarf:** 12 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Schriftzeichen, Rechenmaschine, Computer</b>                      (a) Anhand von Schwerpunkten, wie z.B. Datenspeicherung, Maschinen, Vernetzung sollen wichtige Entwicklungen der Informatik vorgestellt werden.                      (b) Anhand der unterschiedlichen Schwerpunkte sollen universelle Tendenzen der Entwicklung der Informationsverarbeitung erarbeitet werden.</p> <p><b>2. Die Informationsverarbeitung und ihre Möglichkeiten und Gefahren</b>                      (a) Ausgehend von 1. werden Tendenzen der Entwicklung der Informatik erarbeitet                      (b) Informatik wird als Hilfswissenschaft klassifiziert, die weit über ihren originären Bereich hinaus Effizienz- und Leistungssteigerungen erzeugt                      (c) Anhand von Fallbeispielen</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>- nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K)</li> </ul>	<p><i>Die digitale Welt 100 – Von der Schrift zum Smartphone</i></p> <p><i>Die digitale Welt 100 – Das Leben in der digitalen Welt</i></p>

werden technische und organisatorische Vorteile, sowie deren datenschutzrechtlichen Nachteile betrachtet.		
---	--	--

## 2.2.2 Qualifikationsphase 1 - Grundkurs

Die folgende Kompetenz wird in allen Unterrichtsvorhaben der Qualifikationsphase vertieft und soll aus Gründen der Lesbarkeit nicht in jedem Unterrichtsvorhaben separat aufgeführt werden:

Die Schülerinnen und Schüler ...

- ... nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung von Dateien unter Berücksichtigung der Rechteverwaltung (K).

### 2.2.2.1 Unterrichtsvorhaben GK-Q1-I

**Thema:** Wiederholung und Vertiefung der objektorientierten Modellierung

**Leitfragen:** Wie wird aus einem anwendungsbezogenen Sachkontext ein informatisches Klassenmodell entwickelt? Wie werden Attribute, Methoden und Beziehungen identifiziert, den Klassen zugeordnet und dargestellt? Welche Auswirkungen hat die informatisch-technische Entwicklung auf das Leben der Menschen?

**Vorhabenbezogene Konkretisierung:**

Der bereits bekannte objektorientierte Zugang zu informatischer Modellierung wird von einer allgemeinen Betrachtung dieses informatischen Konzepts auf eine konkrete Problematik übertragen. Anhand dieser wird eine anwendungsbezogene Implementation Schritt für Schritt von der Objektidentifikation über das Entwurfs- und Implementationsdiagramm durchlaufen.

Grundlegende Modellierungskonzepte wie Sichtbarkeiten, Assoziationen, Vererbung sowie deren Darstellung in Entwurfs- und Klassendiagrammen und Dokumentationen werden wiederholt. Ebenso wird erneut die grafische Darstellung von Objektkommunikation thematisiert.

Anhand von Gütekriterien und Eigenschaften von Modellierung entwickeln und bewerten die Schülerinnen und Schüler Klassenentwürfe.

Das Konzept der objektorientierten Modellierung wird um die Idee der abstrakten Klasse sowie um das Subtyping erweitert.

**Zeitbedarf:** ca. 14 Stunden

**Sequenzierung des Unterrichtsvorhabens:** s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Mögliche Unterrichtsideen und Materialien
<p><b>1. Wiederholung der grundlegenden Konzepte der objektorientierten Programmierung</b></p> <p>a) Sichtweise der objektorientierten Informatik auf die Welt</p> <p>b) OOP als informatikspezifische Modellierung der Realität</p> <p>c) Schritte der Softwareentwicklung</p>	<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 (M),</li> <li>- analysieren und erläutern objektorientierte Modellierungen (A),</li> <li>- modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M),</li> </ul>	<p><b>Kapitel 1 Konzepte des objektorientierten Modellierens</b></p> <p>1.1 Modellierung der Realität</p> <p>1.2 Die Welt ist voller Objekte Projekteinstieg: Klassentwurf – step by step</p>
<p><b>2. Erweiterung der objektorientierten Programmierung</b></p> <p>a) Umsetzung einer Anforderung in Entwurfs- und Klassendiagramm</p> <p>b) Objektkommunikation im Sequenzdiagramm</p> <p>c) Klassendokumentation</p> <p>d) Umsetzung von Teilen der Modellierung</p>	<ul style="list-style-type: none"> <li>- ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche 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 Möglichkeiten der Polymorphie</li> </ul>	<p>1.3 Gut geplant – Klassentwurf</p> <p>1.4 Hierarchien machen's einfacher – Vererbung</p>
<p><b>3. Mensch und Technik</b></p> <p>a) Informatiker verändern die Welt</p> <p>b) Automatisierung des Alltags durch Informatik</p> <p><b>4. Einführung in die Rekursion</b></p> <p>a) Idee der Rekursion</p> <p>b) Rekursive Grafiken</p> <p>c) Optional: Türme von Hanoi</p> <p>d) Java-Klasse BigInteger</p>	<ul style="list-style-type: none"> <li>- nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> <li>- wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I),</li> <li>- stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D),</li> <li>- dokumentieren Klassen (D),</li> <li>- stellen die Kommunikation zwischen Objekten grafisch dar (D)</li> </ul>	<p>Die digitale Welt 001 – Mensch und Technik</p> <p>Ideen aus B. Schriek (Informatik 2)</p>

### 2.2.2.2 Unterrichtsvorhaben GK-Q1-II

**Thema:**

Organisation und Verarbeitung von Daten I – Modellierung und Implementierung von Anwendungen mit dynamischen und linearen Datenstrukturen

**Leitfragen:**

Wie müssen Daten linear strukturiert werden, um in den gestellten Anwendungsszenarien eine beliebige Anzahl von Objekten verwalten zu können?

**Vorhabensbezogene Konkretisierung:**

Ausgehend von Alltagsbeispielen werden als Erstes die Anforderungen an eine Datenstruktur erschlossen. Anschließend werden die Möglichkeiten des Arrays untersucht, lineare Daten zu verwalten und über deren Grenzen/Probleme die Vorteile einer dynamischen linearen Struktur am Beispiel der Struktur Queue erarbeitet (Anwendungskontext Warteschlange). Der Aufbau der Klasse *Queue* wird vorgegeben und die Methoden werden im Kurs entwickelt und implementiert. Zur Vertiefung der Kenntnisse wird ein weiteres Anwendungsszenario eingeführt, dessen Lösung modelliert und implementiert wird. Darauf folgt die Erarbeitung der Struktur Stack, die mithilfe eines einfachen Anwendungsszenarios eingeführt wird. Auch hier wird die Klasse *Stack* selbst vorgegeben und die Operationen erläutert. Eine Implementierung im Grundkurs ist nicht vorgesehen, kann aber optional durchgeführt werden. Weitere Aufgaben dienen der Vertiefung und Sicherung.

Als letzte lineare dynamische Datenstruktur wird die Liste eingeführt. In dieser Sequenz liegt der Fokus auf der Möglichkeit, auf jedes Element zugreifen zu können. Nachdem die umfangreicheren Standardoperationen dieser Datenstruktur in einem einführenden Beispiel erarbeitet und in einem weiteren Beispiel vertieft wurden, werden abschließend in einem Anwendungskontext verschiedene lineare Datenstrukturen angewendet. Die Modellierung erfolgt beim gesamten Vorhaben in Entwurfs- und Implementationsdiagrammen.

**Zeitbedarf:** 20 Stunden

**Sequenzierung des Unterrichtsvorhabens:** s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Die Datenstruktur Feld</b></p> <p>a) Erarbeitung der Anforderungen an eine Datenstruktur</p> <p>b) Wiederholung der Datenstruktur Array, Eigenschaften der Datenstruktur, Standardoperationen für ein und zwei-dimensionale Arrays</p> <p>c) Modellierung und Implementierung von Anwendungen</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> <li>- erläutern Operationen dynamischer (linearer) Datenstrukturen (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> </ul>	<p><b>Kapitel 2 Lineare Datenstrukturen</b></p> <p>2.1 Anforderungen an eine Datenstruktur</p> <p>2.2 Datenansammlungen fester Größe – Arrays</p>
<p><b>2. Die Datenstruktur Schlange</b></p> <p>a) Modellierung und Implementierung der Verknüpfung von Objekten</p> <p>b) Generische Typen, Trennung von Verwaltung und Inhalt dyn. DS.</p> <p>c) Erläuterung von Problemstellungen, die nach dem FIFO-Prinzip bearbeitet werden</p> <p>d) Funktionalität der Schlange unter Verwendung der Klasse <i>Queue</i>; Erschließen der Standardoperationen</p> <p>e) Modellierung und Implementierung einer Anwendung auf der Basis einer Anforderungsbeschreibung mit Objekten der Klasse <i>Queue</i></p>	<ul style="list-style-type: none"> <li>- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>- stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D)</li> <li>- modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M)</li> <li>- ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M)</li> <li>- dokumentieren Klassen (D)</li> <li>- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I)</li> <li>- verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie (M)</li> </ul>	<p>2.3 Wer zuerst kommt ... – Schlangen</p>
<p><b>3. Die Datenstruktur Stapel</b></p> <p>a) Erläuterung von Problemstellungen, die nach dem LIFO-Prinzip bearbeitet werden</p> <p>b) Funktionalität der Klasse Stapel unter Verwendung der Klasse <i>Stack</i>, Erschließen der Standardoperationen</p> <p>c) Modellierung und Implementierung einer Anwendung auf Basis einer Anforderungsbeschreibung mit Objekten der Klasse <i>Stack</i></p> <p>d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (Objekte der Klassen <i>Queue</i>, <i>Stack</i> und <i>Array (Palindrom)</i>)</p>		<p>2.4 Daten gut abgelegt – Stapel</p>



<p><b>4. Die Datenstruktur Liste</b></p> <p>a) Analyse der Möglichkeiten bisheriger Datenstrukturen zwecks Bestimmung notwendiger Funktionalitäten für komplexere Anwendungen (Abgrenzung zu <i>Stack/Queue</i>, zusätzliche Fähigkeiten der Klasse <i>List</i>)</p> <p>b) Erarbeitung der Funktionalität der Liste unter Verwendung der Klasse <i>List</i></p> <p>c) Modellierung und Implementierung einer Anwendung mit Objekten der Klasse <i>List</i></p> <p>d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (<i>Stack</i>, <i>Queue</i>, <i>List</i>)</p>		<p>2.5 Flexibel für alle Fälle – lineare Listen</p>
<p><b>5. Übungen und Vertiefungen zur Verwendung linearer und dynamischer Datenstrukturen anhand weiterer Problemstellungen</b></p>		<p>2.6 Prüfungsvorbereitung</p>

### 2.2.2.3 Unterrichtsvorhaben GK-Q1-III

**Thema:** Automaten und formale Sprachen

**Leitfragen:**

Wie lassen sich reale Automaten durch ein Modell formal beschreiben? Wie kann die Art und Weise, wie ein Computer Zeichen (Eingaben) verarbeitet, durch Automaten dargestellt werden? Welche Eigenschaften besitzen Automaten und was können sie leisten? Wie werden sie dargestellt? Wie werden reguläre Sprachen durch eine Grammatik beschrieben? In welchem Verhältnis stehen endliche Automaten und Grammatiken? Welche Anwendungsfälle können durch endliche Automaten und Grammatiken regulärer Sprachen beschrieben werden und welche nicht? Welche Möglichkeiten bieten Informatiksysteme und wo liegen ihre Grenzen?

**Vorhabensbezogene Konkretisierung:**

Ausgehend von der Beschreibung und Untersuchung realer Automaten wird das formale Modell eines endlichen Automaten entwickelt. Neben dem Mealy-Automaten geht es vor allem um den erkennenden endlichen Automaten. Auf die Erarbeitung der Beschreibung folgt die Modellierung eigener Automaten und die Untersuchung bestehender, um die Eigenschaften und Grenzen eines endlichen Automaten zu erkennen. Hierbei wird dessen Verhalten auf bestimmte Eingaben analysiert.

An den Themenkomplex *Endliche Automaten* schließt sich die Erarbeitung von Grammatiken regulärer Sprachen an. Die Untersuchung beginnt bei der Erschließung der formalen Beschreibung und wird mit der Entwicklung von Grammatiken zu regulären Sprachen fortgeführt. Hierbei wird auch die Beziehung von Grammatiken regulärer Sprachen zu endlichen Automaten an Beispielen erarbeitet und analysiert. Hierzu gehört auch die Untersuchung, welche Problemstellungen durch endliche Automaten und reguläre Grammatiken beschrieben werden können und welche nicht.

**Zeitbedarf:** 20 Std.

**Sequenzierung des Unterrichtsvorhabens:** s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Endliche Automaten</b></p> <p>a) Erarbeitung der formalen Beschreibung eines Mealy-Automaten und der Darstellungsformen</p> <p>b) Erarbeitung der formalen Beschreibung eines deterministischen endlichen Automaten (DEA) sowie dessen Darstellungsformen; Erschließung der Fachbegriffe Alphabet, Wort, (akzeptierte) Sprache, Determinismus</p> <p>c) Analyse der Eigenschaften von DEAs durch die Modellierung eines Automaten zu einer gegebenen Problemstellung, der Modifikation eines Automaten sowie die Überführung der gegebenen Darstellungsform in eine andere</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> <li>- analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens bei bestimmten Eingaben (A),</li> <li>- ermitteln die Sprache, die ein endlicher Automat akzeptiert (D),</li> <li>- entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M),</li> <li>- stellen endliche Automaten in Tabellen und Graphen dar und überführen sie in die jeweils andere Darstellungsform (D),</li> <li>- entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M),</li> <li>- analysieren und erläutern Grammatiken regulärer Sprachen (A),</li> <li>- modifizieren Grammatiken regulärer Sprachen (M),</li> </ul>	<p><b>Kapitel 4 Endliche Automaten und formale Sprachen</b></p> <p>4.1 Endliche Automaten Projekteinstieg: Schatzsuche</p>
<p><b>2. Grammatiken regulärer Sprachen</b></p> <p>a) Erarbeitung der formalen Beschreibung einer regulären Grammatik (Sprache, Terminal und Nicht-Terminal, Produktionen und Produktionsvorschriften)</p> <p>b) Analyse der Eigenschaften einer regulären Grammatik durch deren Entwicklung und Modellierung zu einer gegebenen Problemstellung.</p>	<ul style="list-style-type: none"> <li>- ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A),</li> <li>- entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M)</li> <li>- entwickeln zur akzeptierten Sprache eines Automaten eine zugehörige Grammatik (M),</li> <li>- beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D),</li> </ul>	<p>4.2 Formale Sprachen</p>
<p><b>3. Übungen und Vertiefungen</b></p> <p>Verwendung endlicher Automaten und Grammatiken regulärer Sprachen</p>	<ul style="list-style-type: none"> <li>- zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A),</li> <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>	<p>4.5 Prüfungsvorbereitung</p>
<p><b>4. Grundsätzliche Arbeitsweise eines Computers und Grenzen der Berechenbarkeit</b></p> <p>a) Grenzen der Berechenbarkeit anhand des Halteproblems, nicht effizient berechenbare Probleme</p>		<p>Die digitale Welt 101 – Maschinennahe Programmierung</p> <p>Die digitale Welt 100 – Berechenbarkeit</p>

#### 2.2.2.4 Unterrichtsvorhaben GK-Q1-IV

**Thema:** Organisation und Verarbeitung von Daten II – Modellierung und Implementierung von Anwendungen mit dynamischen nicht-linearen Datenstrukturen

**Leitfragen:** Wie können Daten mithilfe von Baumstrukturen verwaltet werden? Wie können mit binären Suchbäumen Inhalte sortiert verwaltet werden und welche Vor- und Nachteile bietet dies?

**Vorhabenbezogene Konkretisierung:**

Anhand des Anwendungskontextes Spielbäume werden zunächst der generelle Aufbau von Baumstrukturen (auch nicht-binäre) und wichtige Grundbegriffe erarbeitet. Die Darstellung von Bäumen mit Knoten und Kanten wird eingeführt.

Anschließend rückt der Fokus auf die binären Bäume, deren rekursiver Aufbau für die Traversierung der Datenstruktur genutzt wird. Die Preorder-Traversierung wird verwendet, um einen gespeicherten Inhalt in einem Binärbaum zu finden (Tiefensuche).

Der Anwendungskontext Ahnenbaum wird mithilfe der Klasse *BinaryTree* (der Materialien für das Zentralabitur in NRW) modelliert und (ggf. in Teilen) implementiert. Dabei wird u. a. die Erzeugung eines Binärbaums mithilfe der beiden Konstruktoren der Klasse *BinaryTree* thematisiert.

Möchte man Daten geordnet speichern, bietet sich die Struktur des binären Suchbaums an. An Beispielen wird zunächst das Prinzip des binären Suchbaums erarbeitet. Die Operationen des Suchens, Einfügens, Löschens und der sortierten Ausgabe werden thematisiert.

Um Daten in einem Anwendungskontext mithilfe eines binären Suchbaums verwalten zu können, müssen sie in eine Ordnung gebracht werden können, d. h. sie müssen vergleichbar sein. Diese Vorgabe wird mithilfe des Interfaces *ComparableContent* realisiert, das alle Klassen, dessen Objekte in einem Suchbaum verwaltet werden sollen, implementieren müssen. Auf diese Weise wird ein Anwendungskontext mithilfe der Klassen *BinarySearchTree* und *ComparableContent* modelliert und (ggf. in Teilen) implementiert.

**Zeitbedarf:** 20 Stunden

**Sequenzierung des Unterrichtsvorhabens:** s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Aufbau von Baumstrukturen und Grundbegriffe</b></p> <p>a) Erarbeitung der Begriffe Wurzel, Knoten, Blatt, Kante, Grad eines Knotens und eines Baumes, Pfad, Tiefe, Ebene, Teilbaum</p> <p>b) Aufbau und Darstellung von Baumstrukturen in verschiedenen Anwendungskontexten</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> <li>- stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D),</li> <li>- erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A),</li> <li>- analysieren und erläutern Algorithmen und Programme (A),</li> <li>- stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).</li> </ul>	<p><b>Kapitel 5 Nicht-lineare Datenstrukturen</b></p> <p>5.1 Spielen mit Struktur – Baumstrukturen</p> <p>Projekteinstieg 1: Spielbäume</p>
<p><b>2. Binäre Bäume</b></p> <p>a) rekursiver Aufbau eines binären Baums</p> <p>b) Traversierungen (pre-, in-, postorder)</p> <p>c) Modellierung eines Binärbaums in einem Anwendungskontext mit Hilfe der Klasse BinaryTree (als Entwurfs- und Implementationsdiagramm)</p> <p>d) Implementation einer Anwendung der Datenstruktur binärer Baum (ggf. in Teilen)</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> </ul>	<p>5.2 Zwei Nachfolger sind genug! - Binäre Bäume</p> <p>Implementation des Projekts Ahnenbaum</p>
<p><b>3. Binäre Suchbäume</b></p> <p>a) Prinzip des binären Suchbaums, Ordnungsrelation</p> <p>b) Operationen auf dem binären Suchbaum (Suchen, Einfügen, Löschen, sortierte Ausgabe)</p> <p>c) Modellierung eines binären Suchbaums in einem Anwendungskontext mit Hilfe der Klasse BinarySearchTree (als Entwurfs- und Implementationsdiagramm) und dem Interface Item</p> <p>d) Implementation einer Anwendung der Datenstruktur binärer Suchbaum (ggf. in Teilen)</p>	<ul style="list-style-type: none"> <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> </ul>	<p>5.3 Wer Ordnung hält, spart Zeit beim Suchen – Binäre Suchbäume</p> <p>Projekteinstieg 2: Binäre Suchbäume</p> <p>Implementation des Projekts Benutzerverwaltung</p>
<p><b>4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen</b></p>	<ul style="list-style-type: none"> <li>- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>- testen Programme systematisch anhand von Beispielen (I).</li> </ul>	<p>5.8 Prüfungsvorbereitung</p>

## 2.2.3 Qualifikationsphase 2 – Grundkurs

### 2.2.3.1 Unterrichtsvorhaben GK-Q2-I

**Thema:** Aufbau von Rechnern und Kommunikation in Netzwerken

**Leitfragen:** Was macht menschliche Kommunikation aus? Welchen Stellenwert haben technische/informatische Hilfsmittel für die Kommunikation? Wie werden Daten in einem Netzwerk zwischen den Kommunikationspartnern übertragen? Wie ist die Arbeitsteilung in Netzwerken gestaltet? Wie kann sicher in Netzwerken kommuniziert werden? Was sind die strukturellen Hauptbestandteile eines Computers und wie kann man sich die Ausführung eines maschinennahen Programms mit diesen Komponenten vorstellen?

**Vorhabenbezogene Konkretisierung:**

Ausgehend von alltäglicher Face-to-Face-Kommunikation werden die Grundprinzipien sowie die Bewertungskriterien von Kommunikation erläutert. Das Netzwerk wird als vorteilhafte Kommunikationsstruktur dargestellt und anhand von Topologien und Reichweiten kategorisiert. Ausgehend davon wird der Protokollbegriff entwickelt und anhand des TCP/IP-Schichtenmodells analysiert. Anschließend wird das Client-Server-Prinzip vorgestellt und angewandt.

Sichere Kommunikation in Netzen ist nur dank kryptografischer Verfahren möglich. Stellvertretend werden zwei symmetrische und ein asymmetrisches Verfahren erläutert, angewandt und bewertet.

**Zeitbedarf:** 12 Stunden

**Sequenzierung des Unterrichtsvorhabens:** s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Technische Kommunikation als Fortführung natürlicher Kommunikation</b></p> <p>a) Kommunikation im Shannon-Weaver-Modell</p> <p>b) Kriterien von technischen Kommunikationsarten</p> <p>c) Die Geschichte der technischen Kommunikation</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> <li>- beschreiben und erläutern Netzwerk-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> </ul>	<p><b>Kapitel 6 Kommunikation in Netzwerken</b></p> <p>6.1 Menschen kommunizieren – ohne und mit Technik</p> <p>Projekteinstieg: Kommunikation im Wilden Westen</p>
<p><b>2. Aufbau von Netzwerken und Kommunikationsregeln</b></p> <p>a) Das Netzwerk als Organisationsprinzip der Kommunikation und Möglichkeiten der Ausformung</p> <p>b) Geregelte technische Kommunikation durch Protokolle in Schichtenmodellen</p>	<ul style="list-style-type: none"> <li>- nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zur Erschließung, Aufbereitung und Präsentation fachlicher Inhalte (D).</li> </ul>	<p>6.2 Ohne Protokoll läuft nichts – Netzwerke</p>
<p><b>3. Aufgabenteilung in Netzwerken durch Server und Client</b></p> <p>a) Aufbau und Aufgaben der Client-Server-Struktur</p> <p>b) Protokolle zwischen Client und Server</p>		<p>6.3 Einer für alle – Client-Server-Struktur</p>
<p><b>4. Kryptologie</b></p> <p>a) Veranschaulichen und Anwenden von symmetrischen und asymmetrischen kryptographischen Verfahren (Caesar, Vigenère, RSA)</p> <p>b) Bewertung der Verfahren hinsichtlich ihrer Sicherheit und ihrem Aufwand</p>		<p>Die digitale Welt 111 – Kryptologie</p>
<p><b>5. Grundsätzliche Arbeitsweise eines Computers</b></p> <p>a) Von-Neumann-Architektur und die Ausführung maschinennaher Programme</p> <p><b>6. Übung und Vertiefung des Aufbaus von und der Kommunikation in Netzwerken</b></p>		<p>6.4 Prüfungsvorbereitung</p>

### 2.2.3.2 Unterrichtsvorhaben GK-Q2-II

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

**Leitfragen:** Was sind Datenbanken und wie kann man mit ihnen arbeiten? Wie entwickelt man selbst eine Datenbank für einen Anwendungskontext?

**Vorhabenbezogene Konkretisierung:**

An einem Einführungsbeispiel wird der Aufbau einer Datenbank sowie wichtige Grundbegriffe erarbeitet. Die Schülerinnen und Schüler nehmen dabei zunächst die Sicht der Anwender ein, die eine bestehende Datenbank beschreiben und analysieren und mithilfe von SQL-Abfragen Daten gezielt herausfiltern.

Mithilfe des Projekteinstiegs „Tabellen“ können bereits zu einem frühen Zeitpunkt des Unterrichtsvorhabens Redundanzen, Inkonsistenzen und Anomalien problematisiert werden. Nachdem die Lernenden in der ersten Sequenz mit Datenbanken vertraut gemacht wurden, nehmen sie nun die Rolle der Entwickler an, indem sie selbst Datenbanken von Grund auf modellieren und das Modell in ein Relationenschema überführen. Sie arbeiten mit Entity-Relationship-Diagrammen, um Entitäten, zugehörige Attribute, Relationen und Kardinalitäten in Anwendungskontexten darzustellen. Gegebene ER-Diagramme werden analysiert, erläutert und modifiziert.

Der bereits in der ersten Sequenz problematisierte Begriff der Redundanz wird am Ende des Unterrichtsvorhabens wieder aufgegriffen, um die Normalisierung von Datenbanken zu thematisieren. Bestehende Datenbankschemata werden hinsichtlich der 1. bis 3. Normalform untersucht und (soweit nötig) normalisiert.

**Zeitbedarf:** 20 Stunden

**Sequenzierung des Unterrichtsvorhabens:**



Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Nutzung von relationalen Datenbanken</b></p> <p>a) Aufbau von Datenbanksystemen und Grundbegriffe</p> <ul style="list-style-type: none"> <li>- Aufgaben und Eigenschaften eines Datenbanksystems</li> <li>- Erarbeitung der Begriffe Tabelle, Attribut, Attributwert, Datensatz, Datentyp, Primärschlüssel, Datenbankschema</li> <li>- Problematisierung von Redundanzen, Anomalien und Inkonsistenzen</li> </ul> <p>b) SQL-Abfragen</p> <ul style="list-style-type: none"> <li>- Erarbeitung der grundlegenden Sprachelemente von SQL (SELECT(DISTINCT), FROM, WHERE, JOIN)</li> <li>- Analyse und Erarbeitung von SQL-Abfragen (AND, OR, NOT, 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, geschachtelte Select-Ausdrücke)</li> </ul> <p>c) Vertiefung an einem weiteren Datenbankbeispiel</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> <li>- erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A),</li> <li>- analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A),</li> <li>- verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I),</li> <li>- ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D),</li> <li>- ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M),</li> <li>- stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D),</li> <li>- modifizieren eine Datenbankmodellierung (M),</li> <li>- modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M),</li> <li>- bestimmen Primär- und Sekundärschlüssel (M),</li> <li>- analysieren und erläutern eine Datenbankmodellierung (A),</li> <li>- erläutern die Eigenschaften normalisierter Datenbankschemata (A),</li> </ul>	<p><b>Kapitel 8 Datenbanken</b></p> <p>7.1 Wissen speichern und verwalten – Datenbanksysteme</p> <p>7.2 Daten anordnen mit Tabellen</p> <p>Projekteinstieg: Tabellen</p> <p>7.3 Daten filtern mit SQL</p> <p>7.4 Komplexe Filter</p>
<p><b>2. Modellierung von relationalen Datenbanken</b></p> <p>a) Datenbankentwurf durch ER-Diagramme</p> <ul style="list-style-type: none"> <li>- Ermittlung von Entitäten, zugehörigen Attributen, Beziehungen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms</li> <li>- Erläuterung und Erweiterung einer Datenbankmodellierung</li> </ul> <p>b) Entwicklung eines relationalen Modells aus einem Datenbankentwurf</p> <ul style="list-style-type: none"> <li>- Überführung eines Entity-Relationship-Diagramms in ein relationales Datenbankschema inklusive der Bestimmung von</li> </ul>	<ul style="list-style-type: none"> <li>- überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D).</li> <li>-überführen Datenbankschemata in die 1. bis 3. Normalform (M).</li> </ul>	<p>7.5 Datenbankentwurf</p> <p>7.6 Umsetzung des ER-Modells</p> <p>Wiederaufgriff des Projekteinstiegs</p> <p>7.7 Datenbanken verbessern durch Normalformen</p>

<p>Primär- und Fremdschlüsseln</p> <p>c) Normalformen</p> <p>- Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten)</p>		
<p><b>3. Übung und Vertiefung der Nutzung und Modellierung von relationalen Datenbanken</b></p>		<p>7.8 Prüfungsvorbereitung</p>

### 2.2.3.3 Unterrichtsvorhaben GK-Q2-III

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

**Leitfragen:** Nach welchen Grundprinzipien können Algorithmen strukturiert werden? Welche Qualitätseigenschaften sollten Algorithmen erfüllen? Wie können mithilfe von Such- und Sortieralgorithmen Daten in linearen Strukturen effizient (wieder-)gefunden werden?

**Vorhabenbezogene Konkretisierung:**

Zunächst werden anhand eines Anwendungsbeispiels übergreifende Algorithmeigenschaften (wie Korrektheit, Effizienz und Verständlichkeit) erarbeitet und Schritte der Algorithmenentwicklung wiederholt. Dabei kommen Struktogramme zur Darstellung von Algorithmen zum Einsatz.

Als besondere Struktur von Algorithmen wird die Rekursion an Beispielen wiederholt und gegenüber der Iteration abgegrenzt. Rekursive Algorithmen werden von den Schülerinnen und Schülern analysiert und selbst entwickelt.

In der zweiten Unterrichtssequenz geht es um die Frage, wie Daten in linearen Strukturen (lineare Liste und Array) (wieder-)gefunden werden können. Die lineare Suche als iteratives und die binäre Suche als rekursives Verfahren werden veranschaulicht und implementiert. Die Bewertung der Algorithmen erfolgt, indem jeweils die Anzahl der Vergleichsoperationen und der Speicherbedarf ermittelt wird. Möchte man Daten effizient in einer linearen Struktur wiederfinden, so rückt zwangsläufig die Frage nach einer Sortierstrategie in den Fokus. Es wird mindestens ein iteratives und ein rekursives Sortierverfahren erarbeitet und implementiert sowie ihre Effizienz bewertet.

**Zeitbedarf:** 20 Stunden

**Sequenzierung des Unterrichtsvorhabens:** s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Eigenschaften von Algorithmen</b></p> <p>a) Qualitätseigenschaften von Algorithmen</p> <p>b) Strukturierung von Algorithmen mit Hilfe der Strategien „Modularisierung“ und „Teile und Herrsche“</p> <p>c) Analyse und Entwicklung von rekursiven Algorithmen</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> <li>- analysieren und erläutern Algorithmen und Programme (A),</li> <li>- modifizieren Algorithmen und Programme (I),</li> <li>- stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D),</li> <li>- entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M),</li> </ul>	<p><b>Kapitel 3 Algorithmen</b></p> <p>3.1 Ohne Algorithmen läuft nichts</p> <p>3.2 Teile die Arbeit – rekursive Algorithmen</p>
<p><b>2. Suchen in Listen und Arrays</b></p> <p>a) Lineare Suche in Listen und Arrays</p> <p>b) Binäre Suche in einem Array</p> <p>c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf</p>	<ul style="list-style-type: none"> <li>- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),</li> <li>- testen Programme systematisch anhand von Beispielen (I),</li> <li>- implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I),</li> <li>- beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A),</li> </ul>	<p>3.3 Suchen – iterativ und rekursiv</p> <p>Projekteinstieg: Laufzeitanalyse experimentell</p>
<p><b>3. Sortieren auf Listen und Arrays</b></p> <p>a) Entwicklung und Implementierung eines iterativen Sortierverfahrens für eine Liste (Sortieren durch Einfügen)</p> <p>b) Entwicklung und Implementierung eines rekursiven Sortierverfahrens für eine Liste (Quicksort)</p> <p>c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf</p> <p>d) Weitere Sortierverfahren auf Listen und Arrays (Sortieren durch Auswählen, Mergesort)</p>	<ul style="list-style-type: none"> <li>- beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</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>- untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes 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> </ul>	<p>3.4 Sortieren – iterativ und rekursiv</p>
<p><b>4. Verantwortung der Informatik</b></p> <p>a) Der Unterschied zwischen Anwender und Produzent von Informatiksystemen</p> <p>b) Informatik und Ethik</p>		<p>Die digitale Welt 011 – Verantwortung der Informatik</p>

### 2.2.3.4 Unterrichtsvorhaben GK-Q2-IV

**Thema:** Informatik und Gesellschaft

**Zeitbedarf:** 4 Stunden

#### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<b>1. Mensch und Technik</b> a) Informatiker verändern die Welt b) Automatisierung des Alltags durch Informatik	Die Schülerinnen und Schüler ...  - untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A),	Die digitale Welt 001 – Mensch und Technik
<b>2. Datenschutz</b> a) Datenschutz als Grundrecht b) Das Datenschutzgesetz c) Datensammler	- 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).	Die digitale Welt 101 – Datenschutz
<b>3. Verantwortung der Informatik</b> a) Der Unterschied zwischen Anwender und Produzent von Informatiksystemen b) Informatik und Ethik		Die digitale Welt 011 – Verantwortung der Informatik

## 2.2.4 Qualifikationsphase 1 - Leistungskurs

### 2.2.4.1 Unterrichtsvorhaben LK-Q1-I

**Thema:** Wiederholung und Vertiefung der objektorientierten Modellierung

**Leitfragen:** Wie wird aus einem anwendungsbezogenen Sachkontext ein informatisches Klassenmodell entwickelt? Wie werden Attribute, Methoden und Beziehungen identifiziert, den Klassen zugeordnet und dargestellt? Welche Auswirkungen hat die informatisch-technische Entwicklung auf das Leben der Menschen?

**Vorhabenbezogene Konkretisierung:**

Der bereits bekannte objektorientierte Zugang zu informatischer Modellierung wird von einer allgemeinen Betrachtung dieses informatischen Konzepts auf eine konkrete Problematik übertragen. Anhand dieser wird eine anwendungsbezogene Implementation Schritt für Schritt von der Objektidentifikation über das Entwurfs- und Implementationsdiagramm durchlaufen.

Grundlegende Modellierungskonzepte wie Sichtbarkeiten, Assoziationen, Vererbung sowie deren Darstellung in Entwurfs- und Klassendiagrammen und Dokumentationen werden wiederholt. Ebenso wird erneut die grafische Darstellung von Objektkommunikation thematisiert.

Anhand von Gütekriterien und Eigenschaften von Modellierung entwickeln und bewerten die Schülerinnen und Schüler Klassenentwürfe.

Das Konzept der objektorientierten Modellierung wird um die Idee der abstrakten Klasse sowie um das Subtyping erweitert.

**Zeitbedarf:** 14 Stunden

**Sequenzierung des Unterrichtsvorhabens:** s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Wiederholung der grundlegenden Konzepte der objektorientierten Programmierung</b></p> <p>a) Sichtweise der objektorientierten Informatik auf die Welt</p> <p>b) OOP als informatikspezifische Modellierung der Realität</p> <p>c) Schritte der Softwareentwicklung</p>	<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 (M),</li> <li>- analysieren und erläutern objektorientierte Modellierungen (A),</li> <li>- modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M),</li> </ul>	<p><b>Kapitel 1 Konzepte des objektorientierten Modellierens</b></p> <p>1.1 Modellierung der Realität</p> <p>1.2 Die Welt ist voller Objekte Projekteinstieg: Klassentwurf – step by step</p>
<p><b>2. Erweiterung der objektorientierten Programmierung</b></p> <p>a) Umsetzung einer Anforderung in Entwurfs- und Klassendiagramm</p> <p>b) Objektkommunikation im Sequenzdiagramm</p> <p>c) Klassendokumentation</p> <p>d) Umsetzung von Teilen der Modellierung</p>	<ul style="list-style-type: none"> <li>- ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche 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 Möglichkeiten der Polymorphie (M),</li> </ul>	<p>1.3 Gut geplant – Klassentwurf</p> <p>1.4 Hierarchien machen's einfacher – Vererbung</p>
<p><b>3. Mensch und Technik</b></p> <p>a) Informatiker verändern die Welt</p> <p>b) Automatisierung des Alltags durch Informatik</p>	<ul style="list-style-type: none"> <li>- nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> <li>- wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I),</li> <li>- stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D),</li> <li>- dokumentieren Klassen (D),</li> <li>- stellen die Kommunikation zwischen Objekten grafisch dar (D),</li> <li>- untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes 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> </ul>	<p>Die digitale Welt 001 – Mensch und Technik</p>

#### 2.2.4.2 Unterrichtsvorhaben LK-Q1-II

**Thema:**

Organisation und Verarbeitung von Daten I – Modellierung und Implementierung von Anwendungen mit dynamischen und linearen Datenstrukturen

**Leitfragen:**

Wie müssen Daten linear strukturiert werden, um in den gestellten Anwendungsszenarien eine beliebige Anzahl von Objekten verwalten zu können?

**Vorhabensbezogene Konkretisierung:**

Ausgehend von Alltagsbeispielen werden als Erstes die Anforderungen an eine Datenstruktur erschlossen. Anschließend werden die Möglichkeiten des Arrays untersucht, lineare Daten zu verwalten und über deren Grenzen/Probleme die Vorteile einer dynamischen linearen Struktur am Beispiel der Struktur Queue erarbeitet (Anwendungskontext Warteschlange). Die Klasse *Queue* selbst wird vorgegeben, die Operationen erläutert. Zur Vertiefung der Kenntnisse wird ein weiteres Anwendungsszenario eingeführt (Polizeikontrolle), dessen Lösung modelliert und implementiert wird. Darauf folgt die Erarbeitung der Struktur Stack, die mithilfe eines einfachen Anwendungsszenarios eingeführt (Biber/Palindrom) wird. Auch hier wird die Klasse *Stack* selbst vorgegeben und die Operationen erläutert. Weitere Aufgaben dienen der Vertiefung und Sicherung.

Um die Unterschiede der beiden Prinzipien FIFO und LIFO zu verstehen, werden zur Lösung der Aufgaben sowohl der Stack als auch die Queue benötigt.

Als letzte lineare dynamische Datenstruktur wird die Liste eingeführt. In dieser Sequenz liegt der Fokus auf der Möglichkeit, auf jedes Element zugreifen zu können. Nachdem die umfangreicheren Standardoperationen dieser Datenstruktur in einem einführenden Beispiel (Vokabeltrainer) erarbeitet und in einem weiteren Beispiel vertieft (LED) wurden, werden abschließend in einem Anwendungskontext verschiedene lineare Datenstrukturen angewendet. Die Modellierung erfolgt beim gesamten Vorhaben in Entwurfs- und Implementationsdiagrammen.

**Zeitbedarf:** 30 Stunden

**Sequenzierung des Unterrichtsvorhabens:** s. nächste Seite



Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Die Datenstruktur Feld</b></p> <p>a) Erarbeitung der Anforderungen an eine Datenstruktur</p> <p>b) Wiederholung der Datenstruktur Array, Eigenschaften der Datenstruktur, Standardoperationen für ein und zwei-dimensionale Arrays</p> <p>c) Modellierung und Implementierung von Anwendungen</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> <li>- erläutern Operationen dynamischer (linearer) Datenstrukturen (A),</li> <li>- implementieren Operationen dynamischer (linearer oder nichtlinearer) Datenstrukturen (I),</li> <li>- ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M)</li> </ul>	<p><b>Kapitel 2 Lineare Datenstrukturen</b></p> <p>2.1 Anforderungen an eine Datenstruktur</p> <p>2.2 Datenansammlungen fester Größe – Arrays</p>
<p><b>2. Die Datenstruktur Schlange</b></p> <p>a) Modellierung und Implementierung der Verknüpfung von Objekten</p> <p>b) Generische Typen, Trennung von Verwaltung und Inhalt dyn. DS.</p> <p>c) Erläuterung von Problemstellungen, die nach dem FIFO-Prinzip bearbeitet werden</p> <p>d) Funktionalität der Schlange unter Verwendung der Klasse <i>Queue</i>; Erschließen der Standardoperationen</p> <p>e) Modellierung und Implementierung einer Anwendung auf der Basis einer Anforderungsbeschreibung mit Objekten der Klasse <i>Queue</i></p>	<ul style="list-style-type: none"> <li>- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare Datensammlungen zu (M),</li> <li>- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>- stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D),</li> <li>- modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M),</li> <li>- ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M),</li> <li>- dokumentieren Klassen (D),</li> <li>- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),</li> <li>- verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der</li> </ul>	<p>2.3 Wer zuerst kommt... – Schlangen</p>
<p><b>3. Die Datenstruktur Stapel</b></p> <p>a) Erläuterung von Problemstellungen, die nach dem LIFO-Prinzip bearbeitet werden</p> <p>b) Funktionalität der Klasse Stapel unter Verwendung der Klasse <i>Stack</i>, Erschließen der Standardoperationen</p> <p>c) Modellierung und Implementierung einer Anwendung auf Basis einer Anforderungsbeschreibung mit Objekten der Klasse <i>Stack</i></p> <p>d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (Objekte der Klassen <i>Queue</i>, <i>Stack</i> und <i>Array (Palindrom)</i>)</p>	<ul style="list-style-type: none"> <li>- untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes 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> </ul>	<p>2.4 Daten gut abgelegt – Stapel</p>

<p><b>4. Die Datenstruktur Liste</b></p> <p>a) Analyse der Möglichkeiten bisheriger Datenstrukturen zwecks Bestimmung notwendiger Funktionalitäten für komplexere Anwendungen (Abgrenzung zu <i>Stack/Queue</i>, zusätzliche Fähigkeiten der Klasse <i>List</i>)</p> <p>b) Erarbeitung der Funktionalität der Liste unter Verwendung der Klasse <i>List</i></p> <p>c) Modellierung und Implementierung einer Anwendung mit Objekten der Klasse <i>List</i></p> <p>d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (<i>Stack</i>, <i>Queue</i>, <i>List</i>)</p>		<p>2.5 Flexibel für alle Fälle – lineare Listen</p>
<p><b>5. Übungen und Vertiefungen zur Verwendung linearer und dynamischer Datenstrukturen anhand weiterer Problemstellungen</b></p>		<p>2.6 Prüfungsvorbereitung</p>
<p><b>6. Datenschutz</b></p> <p>a) Datenschutz als Grundrecht</p> <p>b) Das Datenschutzgesetz</p> <p>c) Datensammler</p>		<p><i>Projekteinstieg: Wartende Helden</i></p> <p><i>Mit dem Heldenspiel können alle im Kapitel behandelten Datenstrukturen erarbeitet werden. Das Spiel kann bis zu einem beliebigen Grad realisiert werden, sodass es sowohl als Einstieg als auch als ein umfassendes Projekt für lineare Datenstrukturen genutzt werden kann.</i></p> <p>Die digitale Welt 101 – Datenschutz</p>

### 2.2.4.3 Unterrichtsvorhaben LK-Q1-III

**Thema:** Automaten und formale Sprachen

#### **Leitfragen:**

Wie lassen sich reale Automaten durch ein Modell formal beschreiben? Wie kann die Art und Weise, wie ein Computer Zeichen (Eingaben) verarbeitet, durch Automaten dargestellt werden? Welche Eigenschaften besitzen Automaten und was können sie leisten? Wie werden sie dargestellt? Wie werden reguläre Sprachen durch eine Grammatik beschrieben? In welchem Verhältnis stehen endliche Automaten und Grammatiken? Welche Anwendungsfälle können durch endliche Automaten und Grammatiken regulärer Sprachen oder durch Kellerautomaten und kontextfreie Grammatiken beschrieben werden und welche nicht? Was sind die strukturellen Hauptbestandteile eines Computers und wie kann man sich die Ausführung eines maschinenahen Programms mit diesen Komponenten vorstellen? Welche Möglichkeiten bieten Informatiksysteme und wo liegen ihre Grenzen? In welchen Anwendungen finden Automaten Einsatz?

#### **Vorhabensbezogene Konkretisierung:**

Ausgehend von der Beschreibung und Untersuchung realer Automaten wird das formale Modell eines endlichen Automaten entwickelt. Neben dem Mealy-Automaten geht es vor allem um den erkennenden endlichen Automaten. Auf die Erarbeitung der Beschreibung folgt die Modellierung eigener Automaten und die Untersuchung bestehender, um die Eigenschaften und Grenzen eines endlichen Automaten zu erkennen. Hierbei wird dessen Verhalten auf bestimmte Eingaben analysiert. An den Themenkomplex *Endliche Automaten* schließt sich die Erarbeitung von Grammatiken regulärer Sprachen an. Die Untersuchung beginnt bei der Erschließung der formalen Beschreibung und wird mit der Entwicklung von Grammatiken zu regulären Sprachen fortgeführt. Hierbei wird auch die Beziehung von Grammatiken regulärer Sprachen zu endlichen Automaten an Beispielen erarbeitet und analysiert. Hierzu gehört auch die Untersuchung, welche Problemstellungen durch endliche Automaten und reguläre Grammatiken beschrieben werden können und welche nicht. Anhand einer Problemstellung wird die Grenze der endlichen Automaten aufgezeigt und in das Konzept des Kellerautomaten bzw. der kontextfreien Sprachen überführt. Der Anwendungsbezug zu den theoretischen Grundlagen der Sprachen- und Automatentheorie wird durch das Modellieren und Implementieren von Scannern, Parsern und Interpretern vertieft.

**Zeitbedarf:** 40 Std.

#### **Sequenzierung des Unterrichtsvorhabens:**

<b>Unterrichtssequenzen</b>	<b>Zu entwickelnde Kompetenzen</b>	<b>Kapitel und Materialien</b>
<b>1. Endliche Automaten</b> a) Erarbeitung der formalen Beschreibung eines Mealy-Automaten und der Darstellungsformen	Die Schülerinnen und Schüler - analysieren und erläutern die Eigenschaften endlicher Automaten und Kellerautomaten	<b>Kapitel 4 Endliche Automaten und formale Sprachen</b> 4.1. Endliche Automaten Projekteinstieg: Schatzsuche

<p>b) Erarbeitung der formalen Beschreibung eines deterministischen endlichen Automaten (DEA) sowie dessen Darstellungsformen; Erschließung der Fachbegriffe Alphabet, Wort, (akzeptierte) Sprache, Determinismus</p> <p>c) Analyse der Eigenschaften von DEAs durch die Modellierung eines Automaten zu einer gegebenen Problemstellung, der Modifikation eines Automaten sowie die Überführung der gegebenen Darstellungsform in eine andere</p>	<p>einschließlich ihres Verhaltens bei bestimmten Eingaben (A),</p> <ul style="list-style-type: none"> <li>- ermitteln die Sprache, die ein endlicher Automat oder ein Kellerautomat akzeptiert (D),</li> <li>- entwickeln und modifizieren zu einer Problemstellung endliche Automaten oder Kellerautomaten (M),</li> <li>- stellen endliche Automaten in Tabellen und Graphen dar und überführen sie in die jeweils andere Darstellungsform (D),</li> <li>- entwickeln zur Grammatik einer regulären oder kontextfreien Sprache einen zugehörigen endlichen Automaten oder einen Kellerautomaten (M),</li> </ul>	
<p><b>2. Grammatiken regulärer Sprachen</b></p> <p>a) Erarbeitung der formalen Beschreibung einer regulären Grammatik (Sprache, Terminal und Nicht-Terminal, Produktionen und Produktionsvorschriften)</p> <p>b) Analyse der Eigenschaften einer regulären Grammatik durch deren Entwicklung und Modellierung zu einer gegebenen Problemstellung.</p>	<ul style="list-style-type: none"> <li>- analysieren und erläutern Grammatiken regulärer und kontextfreier Sprachen (A),</li> <li>- modifizieren Grammatiken regulärer und kontextfreier Sprachen (M),</li> <li>- ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A),</li> <li>- entwickeln zu einer regulären oder kontextfreien Sprache eine Grammatik, die die Sprache erzeugt (M),</li> </ul>	<p>4.2 Formale Sprachen</p>
<p><b>3. Scanner, Parser, Interpreter</b></p> <p>a) Erarbeitung der Konzepte von Scanner, Parser und Interpreter</p> <p>b) Modellierung und Implementation von Scanner, Parser und Interpreter</p>	<ul style="list-style-type: none"> <li>- entwickeln zur akzeptierten Sprache eines Automaten eine zugehörige Grammatik (M),</li> <li>- beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D),</li> </ul>	<p>4.3 Vom Text zum Programm</p> <p>4.4 Scanner und Parser</p>
<p><b>4. Übungen und Vertiefungen</b></p> <p>Verwendung endlicher Automaten und Grammatiken regulärer Sprachen</p>	<ul style="list-style-type: none"> <li>- erläutern die Grenzen endlicher Automaten und regulärer Sprachen im Anwendungszusammenhang (A),</li> <li>- modellieren und implementieren Scanner, Parser und Interpreter zu einer gegebenen regulären Sprache (I),</li> <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>	<p>4.5 Prüfungsvorbereitung</p>
<p><b>5. Grundsätzliche Arbeitsweise eines Computers und Grenzen der Berechenbarkeit</b></p> <p>a) Von-Neumann-Architektur und die Ausführung maschinennaher</p>		<p>Die digitale Welt 101 – Maschinennahe Programmierung</p> <p>Die digitale Welt 100 –</p>

Programme b) Grenzen der Berechenbarkeit anhand des Halteproblems, nicht effizient berechenbare Probleme		Berechenbarkeit
---	--	-----------------

#### 2.2.4.4 Unterrichtsvorhaben LK-Q1-IV

**Thema:** Organisation und Verarbeitung von Daten II – Modellierung und Implementierung von Anwendungen mit dynamischen nicht-linearen Datenstrukturen

**Leitfragen:** Wie können Daten mithilfe von Baumstrukturen verwaltet werden? Wie können mit binären Suchbäumen Inhalte sortiert verwaltet werden und welche Vor- und Nachteile bietet dies?

**Vorhabenbezogene Konkretisierung:**

Anhand des Anwendungskontextes Spielbäume werden zunächst der generelle Aufbau von Baumstrukturen (auch nicht-binäre) und wichtige Grundbegriffe erarbeitet. Die Darstellung von Bäumen mit Knoten und Kanten wird eingeführt.

Anschließend rückt der Fokus auf die binären Bäume, deren rekursiver Aufbau für die Traversierung der Datenstruktur genutzt wird. Die Preorder-Traversierung wird verwendet, um einen gespeicherten Inhalt in einem Binärbaum zu finden (Tiefensuche).

Der Anwendungskontext Ahnenbaum wird mithilfe der Klasse *BinaryTree* (der Materialien für das Zentralabitur in NRW) modelliert und (ggf. in Teilen) implementiert. Dabei wird u. a. die Erzeugung eines Binärbaums mithilfe der beiden Konstruktoren der Klasse *BinaryTree* thematisiert.

Möchte man Daten geordnet speichern, so bietet sich die Struktur des binären Suchbaums an. An Beispielen wird zunächst das Prinzip des binären Suchbaums erarbeitet. Die Operationen des Suchens, Einfügens, Löschens und der sortierten Ausgabe werden thematisiert.

Um Daten in einem Anwendungskontext mithilfe eines binären Suchbaums verwalten zu können, müssen sie in eine Ordnung gebracht werden können, d. h. sie müssen vergleichbar sein. Diese Vorgabe wird mithilfe des Interfaces *ComparableContent* realisiert, das alle Klassen, dessen Objekte in einem Suchbaum verwaltet werden sollen, implementieren müssen. Auf diese Weise wird ein Anwendungskontext (Benutzerverwaltung) mithilfe der Klassen *BinarySearchTree* und *ComparableContent* modelliert und (ggf. in Teilen) implementiert.

**Zeitbedarf:** 40 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Aufbau von Baumstrukturen und Grundbegriffe</b></p> <p>a) Erarbeitung der Begriffe Wurzel, Knoten, Blatt, Kante, Grad eines Knotens und eines Baumes, Pfad, Tiefe, Ebene, Teilbaum</p> <p>b) Aufbau und Darstellung von Baumstrukturen in verschiedenen Anwendungskontexten</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> <li>- stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D),</li> <li>- erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A),</li> <li>- implementieren Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (I),</li> <li>- analysieren und erläutern Algorithmen und Programme (A),</li> </ul>	<p><b>Kapitel 5 Nicht-lineare Datenstrukturen</b></p> <p>5.1 Spielen mit Struktur – Baumstrukturen</p> <p>Projekteinstieg 1: Spielbäume</p>
<p><b>2. Binäre Bäume</b></p> <p>a) rekursiver Aufbau eines binären Baums</p> <p>b) Traversierungen (pre-, in-, postorder)</p> <p>c) Modellierung eines Binärbaums in einem Anwendungskontext mit Hilfe der Klasse BinaryTree (als Entwurfs- und Implementationsdiagramm)</p> <p>d) Implementation einer Anwendung der Datenstruktur binärer Baum (ggf. in Teilen)</p>	<ul style="list-style-type: none"> <li>- stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).</li> <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</li> </ul>	<p>5.2 Zwei Nachfolger sind genug! - Binäre Bäume</p> <p>Implementation des Projekts Ahnenbaum</p>
<p><b>3. Binäre Suchbäume</b></p> <p>a) Prinzip des binären Suchbaums, Ordnungsrelation</p> <p>b) Operationen auf dem binären Suchbaum (Suchen, Einfügen, Löschen, sortierte Ausgabe)</p> <p>c) Modellierung eines binären Suchbaums in einem Anwendungskontext mit Hilfe der Klasse BinarySearchTree (als Entwurfs- und Implementationsdiagramm) und dem Interface Item</p> <p>d) Implementation einer Anwendung der Datenstruktur binärer Suchbaum (ggf. in Teilen)</p>	<p>Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),</p> <ul style="list-style-type: none"> <li>- verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M),</li> <li>- entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“, „Teilen und Herrschen“ und „Backtracking“ (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> </ul>	<p>5.3 Wer Ordnung hält, spart Zeit beim Suchen – Binäre Suchbäume</p> <p>Projekteinstieg 2: Binäre Suchbäume</p> <p>Implementation des Projekts Benutzerverwaltung</p>
<p><b>4. Graphen</b></p> <p>a) Anwendungskontexte für Graphen, Grundbegriffe</p> <p>b) Modellierung von Graphen mit Adjazenzlisten mit Hilfe der Klassen Graph und GraphNode und Adjazenzmatrizen.</p> <p>c) Algorithmus von Dijkstra</p>	<ul style="list-style-type: none"> <li>- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>- testen Programme systematisch anhand von Beispielen (I).</li> </ul>	<p>Projekteinstieg 3: Navigationssysteme</p> <p>5.4 Navigieren mit Struktur – Graphen</p> <p>5.5 Modellierung von Graphen</p> <p>5.6 Kürzester Weg – der Algorithmus von Dijkstra</p> <p>5.7 Graphen durchsuchen</p>

d) Tiefen- und Breitensuche	
<b>5. Übung und Vertiefungen der Verwendung von Binärbäumen, binären Suchbäumen und Graphen anhand weiterer Problemstellungen</b>	5.8 Prüfungsvorbereitung

## 2.2.5 Qualifikationsphase 2 - Leistungskurs

### 2.2.5.1 Unterrichtsvorhaben LK-Q2-I

**Thema:** Aufbau von und Kommunikation in Netzwerken

**Leitfragen:** Was macht menschliche Kommunikation aus? Welchen Stellenwert haben technische/informatische Hilfsmittel für die Kommunikation? Wie werden Daten in einem Netzwerk zwischen den Kommunikationspartnern übertragen? Wie ist die Arbeitsteilung in Netzwerken gestaltet? Wie kann sicher in Netzwerken kommuniziert werden? Wie werden Protokolle zur zuverlässigen Kommunikation eingesetzt und in Software umgesetzt?

**Vorhabenbezogene Konkretisierung:**

Ausgehend von alltäglicher Face-to-Face-Kommunikation werden die Grundprinzipien sowie die Bewertungskriterien von Kommunikation erläutert. Das Netzwerk wird als vorteilhafte Kommunikationsstruktur dargestellt und anhand von Topologien und Reichweiten kategorisiert. Ausgehend davon wird der Protokollbegriff entwickelt und anhand des TCP/IP-Schichtenmodells analysiert. Anschließend wird das Client-Server-Prinzip vorgestellt und in einer Projektarbeit angewandt.

Es werden eigenständig Protokolle entwickelt und in einer Client-Server-Anwendung umgesetzt.

Sichere Kommunikation in Netzen ist nur dank kryptografischer Verfahren möglich. Stellvertretend werden zwei symmetrische und ein asymmetrisches Verfahren erläutert, angewandt und bewertet.

**Zeitbedarf:** 40 Stunden

**Sequenzierung des Unterrichtsvorhabens:** s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Technische Kommunikation als Fortführung natürlicher Kommunikation</b></p> <p>a) Kommunikation im Shannon-Weaver-Modell</p> <p>b) Kriterien von technischen Kommunikationsarten</p> <p>c) Die Geschichte der technischen Kommunikation</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> <li>- beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A),</li> <li>- analysieren und erläutern Protokolle zur Kommunikation in einem Client-Server-Netzwerk (A),</li> </ul>	<p><b>Kapitel 6 Kommunikation in Netzwerken</b></p> <p>6.1 Menschen kommunizieren – ohne und mit Technik</p> <p>Projekteinstieg: Kommunikation im Wilden Westen</p>
<p><b>2. Aufbau von Netzwerken und Kommunikationsregeln</b></p> <p>a) Das Netzwerk als Organisationsprinzip der Kommunikation und Möglichkeiten der Ausformung</p> <p>b) Geregelte technische Kommunikation durch Protokolle in Schichtenmodellen</p>	<ul style="list-style-type: none"> <li>- entwickeln und erweitern Protokolle zur Kommunikation in einem Client-Server-Netzwerk (M),</li> <li>- erläutern das Prinzip der Nebenläufigkeit (A),</li> <li>- analysieren und erläutern Algorithmen und Methoden zur Client-Server-Kommunikation (A),</li> <li>- entwickeln und implementieren Algorithmen und Methoden zur Client-Server-Kommunikation (I),</li> </ul>	<p>6.2 Ohne Protokoll läuft nichts – Netzwerke</p>
<p><b>3. Aufgabenteilung in Netzwerken durch Server und Client</b></p> <p>a) Aufbau und Aufgaben der Client-Server-Struktur</p> <p>b) Protokolle zwischen Client und Server</p> <p>c) Implementation von Anwendungen unter Berücksichtigung der Client-Server-Struktur und gegebener Protokolle</p>	<ul style="list-style-type: none"> <li>- analysieren und erläutern Eigenschaften, Funktionsweisen und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A),</li> <li>- nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zur Erschließung, Aufbereitung und Präsentation fachlicher Inhalte (D).</li> </ul>	<p>6.3 Einer für alle – Client-Server-Struktur</p> <p>Projektarbeit (z.B. Chatanwendung, Spiele wie Schiffe versenken oder Stein, Schere, Papier, Fussballmanager)</p>
<p><b>4. Kryptologie</b></p> <p>a) Veranschaulichen und Anwenden von symmetrischen und asymmetrischen kryptographischen Verfahren (Caesar, Vigenère, RSA, Diffie-Hellman Schlüsselaustausch)</p> <p>b) Bewertung der Verfahren hinsichtlich ihrer Sicherheit und ihrem Aufwand</p>		<p>Die digitale Welt 111 – Kryptologie</p>
<p><b>5. Übung und Vertiefung des Aufbaus von und der Kommunikation in Netzwerken</b></p>		<p>6.4 Prüfungsvorbereitung</p>



### 2.2.5.2 Unterrichtsvorhaben LK-Q2-II

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

**Leitfragen:** Was sind Datenbanken und wie kann man mit ihnen arbeiten? Wie entwickelt man selbst eine Datenbank für einen Anwendungskontext?

**Vorhabenbezogene Konkretisierung:**

Am Beispiel eines Online-Buchhandels wird der Aufbau einer Datenbank sowie wichtige Grundbegriffe erarbeitet. Die Schülerinnen und Schüler nehmen dabei zunächst die Sicht der Anwender ein, die eine bestehende Datenbank beschreiben und analysieren und mithilfe von SQL-Abfragen Daten gezielt herausfiltern.

Mithilfe des Projekteinstiegs „Tabellen“ können bereits zu einem frühen Zeitpunkt des Unterrichtsvorhabens Redundanzen, Inkonsistenzen und Anomalien problematisiert werden.

Nachdem die Lernenden in der ersten Sequenz mit Datenbanken vertraut gemacht wurden, nehmen sie nun die Rolle der Entwickler an, indem sie selbst Datenbanken von Grund auf modellieren und das Modell in ein Relationenschema überführen. Sie arbeiten mit Entity-Relationship-Diagrammen, um Entitäten, zugehörige Attribute, Relationen und Kardinalitäten in Anwendungskontexten darzustellen. Gegebene ER-Diagramme werden analysiert, erläutert und modifiziert.

Der bereits in der ersten Sequenz problematisierte Begriff der Redundanz wird am Ende des Unterrichtsvorhabens wieder aufgegriffen, um die Normalisierung von Datenbanken zu thematisieren. Bestehende Datenbankschemata werden hinsichtlich der 1. bis 3. Normalform untersucht und (soweit nötig) normalisiert.

Den Abschluss des Unterrichtsvorhabens bildet eine Projektarbeit, bei der eine Datenbankanwendung von den Schülerinnen und Schülern modelliert, implementiert, dokumentiert und getestet wird. Die Implementation der Java-Anwendung wird dabei mit den Klassen *DatabaseConnector* und *QueryResult* (der Materialien für das Zentralabitur NRW) realisiert. Die Projektarbeit vertieft insbesondere die Kompetenzen im Bereich Kommunizieren und Kooperieren, da die Lernenden eigenverantwortlich und kooperativ arbeiten, ihre Arbeitsabläufe bewerten und Ergebnisse vorstellen.

**Zeitbedarf:** 30 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<b>1. Nutzung von relationalen Datenbanken</b> a) Aufbau von Datenbanksystemen und Grundbegriffe - Aufgaben und Eigenschaften eines Datenbanksystems - Erarbeitung der Begriffe Tabelle,	Die Schülerinnen und Schüler ... - 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),	<b>Kapitel 8 Datenbanken</b> 7.1 Wissen speichern und verwalten – Datenbanksysteme 7.2 Daten anordnen mit Tabellen Projekteinstieg: Tabellen

<p>Attribut, Attributwert, Datensatz, Datentyp, Primärschlüssel, Datenbankschema</p> <ul style="list-style-type: none"> <li>- Problematisierung von Redundanzen, Anomalien und Inkonsistenzen</li> </ul> <p>b) SQL-Abfragen</p> <ul style="list-style-type: none"> <li>- Erarbeitung der grundlegenden Sprachelemente von SQL (SELECT(DISTINCT), FROM, WHERE, JOIN)</li> <li>- Analyse und Erarbeitung von SQL-Abfragen (AND, OR, NOT, 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, geschachtelte Select-Ausdrücke)</li> </ul> <p>c) Vertiefung an einem weiteren Datenbankbeispiel</p>	<ul style="list-style-type: none"> <li>- verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I),</li> <li>- ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D),</li> <li>- ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M),</li> <li>- stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D),</li> <li>- modifizieren eine Datenbankmodellierung (M),</li> <li>- modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M),</li> <li>- bestimmen Primär- und Sekundärschlüssel (M),</li> <li>- implementieren eine relationales Datenbankschema als Datenbank (I),</li> <li>- analysieren und erläutern eine Datenbankmodellierung (A),</li> </ul>	<p>7.3 Daten filtern mit SQL</p> <p>7.4 Komplexe Filter</p>
<p><b>2. Modellierung von relationalen Datenbanken</b></p> <p>a) Datenbankentwurf durch ER-Diagramme</p> <ul style="list-style-type: none"> <li>- Ermittlung von Entitäten, zugehörigen Attributen, Beziehungen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms</li> <li>- Erläuterung und Erweiterung einer Datenbankmodellierung</li> </ul> <p>b) Entwicklung eines relationalen Modells aus einem Datenbankentwurf</p> <ul style="list-style-type: none"> <li>- Überführung eines Entity-Relationship-Diagramms in ein relationales Datenbankschema inklusive der Bestimmung von Primär- und Fremdschlüsseln</li> </ul> <p>c) Normalformen</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>	<ul style="list-style-type: none"> <li>- erläutern die Eigenschaften normalisierter Datenbankschemata (A),</li> <li>- überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D),</li> <li>-überführen Datenbankschemata in die 1. bis 3. Normalform (M),</li> <li>- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung von Daten, zur Organisation von Arbeitsabläufen sowie zur Verteilung und Zusammenführung von Arbeitsanteilen (K),</li> <li>- wenden didaktisch orientierte Entwicklungsumgebungen zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I),</li> <li>- entwickeln mit didaktisch orientierten Entwicklungsumgebungen einfache Benutzungsoberflächen zur Kommunikation mit einem Informatiksystem (M).</li> </ul>	<p>7.5 Datenbankentwurf</p> <p>7.6 Umsetzung des ER-Modells</p> <p>Wiederaufgriff des Projekteinstiegs</p> <p>7.7 Datenbanken verbessern durch Normalformen</p>

<b>3. Implementation von relationalen Datenbanken</b> Modellierung, Implementation, Dokumentation und Test einer Datenbankanwendung in einem Anwendungskontext (Projektarbeit)		7.3 Daten filtern mit SQL 7.4 Komplexe Filter 7.6 Umsetzung des ER-Modells Projektarbeit (z.B. Kassensystem, Cocktailverzeichnis, Umfragetool für die Abizeitung), ggf. auch mit Netzwerkfunktionalität
<b>4. Übung und Vertiefung der Nutzung und Modellierung von relationalen Datenbanken</b>		7.8 Prüfungsvorbereitung

### 2.2.5.3 Unterrichtsvorhaben LK-Q2-III

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

**Leitfragen:** Nach welchen Grundprinzipien können Algorithmen strukturiert werden? Welche Qualitätseigenschaften sollten Algorithmen erfüllen? Wie können mithilfe von Such- und Sortieralgorithmen Daten in linearen Strukturen effizient (wieder-)gefunden werden?

**Vorhabenbezogene Konkretisierung:**

Zunächst werden anhand eines Anwendungsbeispiels übergreifende Algorithmeigenschaften (wie Korrektheit, Effizienz und Verständlichkeit) erarbeitet und Schritte der Algorithmenentwicklung wiederholt. Dabei kommen Struktogramme zur Darstellung von Algorithmen zum Einsatz. Als besondere Struktur von Algorithmen wird die Rekursion an Beispielen veranschaulicht und gegenüber der Iteration abgegrenzt. Rekursive Algorithmen werden von den Schülerinnen und Schülern analysiert und selbst entwickelt.

In der zweiten Unterrichtssequenz geht es um die Frage, wie Daten in linearen Strukturen (lineare Liste und Array) (wieder-)gefunden werden können. Die lineare Suche als iteratives und die binäre Suche als rekursives Verfahren werden veranschaulicht und implementiert. Die Bewertung der Algorithmen erfolgt, indem jeweils die Anzahl der Vergleichsoperationen und der Speicherbedarf ermittelt wird. Möchte man Daten effizient in einer linearen Struktur wiederfinden, so rückt zwangsläufig die Frage nach einer Sortierstrategie in den Fokus. Es wird mindestens ein iteratives und ein rekursives Sortierverfahren erarbeitet und implementiert sowie ihre Effizienz bewertet. Hier bietet sich die Implementation einer Testanwendung an, die die Laufzeiten unterschiedlicher Sortieralgorithmen misst. Daran anschließend folgen theoretische Analysen der Laufzeit.

**Zeitbedarf:** 22 Stunden

**Sequenzierung des Unterrichtsvorhabens:** s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Eigenschaften von Algorithmen</b></p> <p>a) Qualitätseigenschaften von Algorithmen</p> <p>b) Strukturierung von Algorithmen mit Hilfe der Strategien „Modularisierung“ und „Teile und Herrsche“</p> <p>c) Analyse und Entwicklung von rekursiven Algorithmen</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> <li>- analysieren und erläutern Algorithmen und Programme (A),</li> <li>- modifizieren Algorithmen und Programme (I),</li> <li>- stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D),</li> <li>- entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M),</li> </ul>	<p><b>Kapitel 3 Algorithmen</b></p> <p>3.1 Ohne Algorithmen läuft nichts</p> <p>3.2 Teile die Arbeit – rekursive Algorithmen</p>
<p><b>2. Suchen in Listen und Arrays</b></p> <p>a) Lineare Suche in Listen und Arrays</p> <p>b) Binäre Suche in einem Array</p> <p>c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf</p>	<ul style="list-style-type: none"> <li>- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),</li> <li>- testen Programme systematisch anhand von Beispielen und mithilfe von Testanwendungen (I),</li> <li>- implementieren und erläutern iterative und rekursive Such- und Sortierverfahren unterschiedlicher Komplexitätsklassen (Speicherbedarf und Laufzeitverhalten) (I),</li> </ul>	<p>3.3 Suchen – iterativ und rekursiv</p> <p>Projekteinstieg: Laufzeitanalyse experimentell</p>
<p><b>3. Sortieren auf Listen und Arrays</b></p> <p>a) Entwicklung und Implementierung eines iterativen Sortierverfahrens für eine Liste (Sortieren durch Einfügen)</p> <p>b) Entwicklung und Implementierung eines rekursiven Sortierverfahrens für eine Liste (Quicksort)</p> <p>c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf (praktische Tests und theoretische Analyse)</p> <p>d) Weitere Sortierverfahren auf Listen und Arrays (Sortieren durch Auswählen, Mergesort)</p> <p><b>4. Verantwortung der Informatik</b></p> <p>a) Der Unterschied zwischen Anwender und Produzent von Informatiksystemen</p> <p>b) Informatik und Ethik</p>	<ul style="list-style-type: none"> <li>- beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A),</li> <li>- beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</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>- untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes 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> </ul>	<p>3.4 Sortieren – iterativ und rekursiv</p> <p>Implementation einer Testanwendung</p> <p>Die digitale Welt 011 – Verantwortung der Informatik</p>

## 2.3 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

In Absprache mit der Lehrerkonferenz sowie unter Berücksichtigung des Schulprogramms hat die Fachkonferenz Informatik die folgenden fachdidaktischen und fachmethodischen Grundsätze beschlossen.

### Überfachliche Grundsätze:

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

### Fachliche Grundsätze:

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

## 2.4 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

Schulische Leistungsbewertung steht im Spannungsfeld pädagogischer und gesellschaftlicher Zielsetzung.

Unter pädagogischen Gesichtspunkten hat sie vornehmlich das Individuum im Blick. Hier soll sie über den Leistungszuwachs rückmelden und dadurch die Motivation für weitere Anstrengungen erhöhen. Sie ermöglicht den Schüler\*innen ihre noch vorhandenen fachlichen Defizite wie auch ihre Stärken und

Fähigkeiten zu erkennen um dadurch ein realistisches Selbstbild aufzubauen. Sie ist Basis für gezielte individuelle Förderung.

Die Fachkonferenz hat im Einklang mit dem entsprechenden schulbezogenen Konzept die nachfolgenden, verbindlichen Grundsätze zur Leistungsbewertung und -rückmeldung beschlossen. Es wird zwischen **schriftlichen** und **sonstigen** Leistungen unterschieden.

### **Grundsätze der Leistungsbewertung**

Die Fachkonferenz Informatik legt die Kriterien für die Leistungsbewertung fest. Die Lehrer\*innen machen diese Kriterien den Schüler\*innen transparent.

Es gelten folgende Grundsätze der Leistungsbewertung:

- Lernerfolgsüberprüfungen sind ein kontinuierlicher Prozess. Bewertet werden alle im Zusammenhang mit dem Unterricht erbrachten Leistungen (schriftliche Arbeiten, mündliche Beiträge, praktische Leistungen).
- Leistungsbewertung bezieht sich auf die im Unterricht geförderten Kompetenzen.
- Die Lehrperson gibt den Schüler\*innen im Unterricht hinreichend Gelegenheit, die entsprechenden Anforderungen der Leistungsbewertung im Unterricht in Umfang und Anspruch kennenzulernen und sich auf sie vorzubereiten.
- Bewertet werden der Umfang, die selbstständige und richtige Anwendung der Kenntnisse, Fähigkeiten und Fertigkeiten sowie die Art der Darstellung.

#### **2.4.1.1 Beurteilungsbereich Klausuren**

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

Die Anzahl und Dauer der schriftlichen Klausuren im Fach Informatik hat die Fachkonferenz wie folgt festgelegt:

Jahrgangsstufe	Arbeiten pro Schuljahr	Dauer (in U-Stunden)
EF	2	2
Q1	4	GK: 2 LK: 4
Q2	4	GK: 3 LK: 4

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

Die Bewertung der schriftlichen Leistungen in Klausuren erfolgt über ein Raster mit Hilfspunkten, die im Erwartungshorizont den einzelnen Kriterien zugeordnet sind.

Die Zuordnung der Punktsommen zu den Notenstufen orientiert sich an dem Zuordnungsschema des Zentralabiturs. Von diesem kann aber im Einzelfall begründet abgewichen werden, wenn z.B. besonders originelle Teillösungen nicht durch Hilfspunkte gemäß den Kriterien des Erwartungshorizontes abbilden lassen oder eine Abwertung wegen besonders schwacher Darstellung angemessen erscheint.

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

#### 2.4.1.2 Beurteilungsbereich „Sonstige Leistungen“

Den Schüler\*innen werden die Kriterien zum Bewertungsbereich sonstige Leistungen zu Beginn des Schuljahres genannt.

Bei der Unterrichtsgestaltung sind den Schüler\*innen hinreichend Möglichkeiten zur Mitarbeit zu eröffnen, z.B. durch

- Beteiligung am Unterrichtsgespräch
- Zusammenfassungen zur Vor- und Nachbereitung des Unterrichts
- Präsentation von Arbeitsergebnissen
- Mitarbeit in Partner- und Gruppenarbeitsphase
- Schriftliche Bearbeitung von Aufgaben im Unterricht
- Implementierung, Test und Anwendung von Informatiksystemen
- Arbeitsmappe und Arbeitstagebuch zu einem durchgeführten Unterrichtsvorhaben
- Lernerfolgsübersichten durch kurze schriftliche Übungen: In Kursen, in denen weniger als 50% der Kursmitglieder eine Klausur schreiben, finden schriftliche Übungen mindestens einmal pro Kurshalbjahr statt, in anderen Kursen entscheidet über die Durchführung die Lehrkraft.
- Protokolle und Referate
- Kürzere Projektarbeiten
- Bearbeitung von schriftlichen Aufgaben im Unterricht.

Der Bewertungsbereich „sonstige Leistungen“ erfasst die Qualität und Kontinuität der Beiträge, die die Schüler\*innen im Unterricht erbringen. Diese Beiträge sollen unterschiedliche mündliche und schriftliche Formen in enger Bindung an die Aufgabenstellung, die inhaltliche Reichweite und das Anspruchsniveau der jeweiligen Unterrichtseinheit umfassen.

#### 2.4.1.3 Bewertungskriterien

Die Bewertungskriterien für eine Leistung müssen auch für Schüler\*innen **transparent, klar und nachvollziehbar** sein. Die folgenden allgemeinen Kriterien gelten sowohl für die schriftlichen als auch für die sonstigen Formen der Leistungsüberprüfung:

- Qualität der Beiträge
- Quantität der Beiträge
- Kontinuität der Beiträge
- Sachliche Richtigkeit
- Angemessene Verwendung der Fachsprache
- Darstellungskompetenz
- Komplexität/Grad der Abstraktion
- Selbstständigkeit im Arbeitsprozess
- Einhaltung gesetzter Fristen
- Präzision
- Differenziertheit der Reflexion
- Bei Gruppenarbeiten
  - Einbringen in die Arbeit der Gruppe
  - Durchführung fachlicher Arbeitsanteile
- Bei Projekten
  - Selbstständige Themenfindung
  - Dokumentation des Arbeitsprozesses
  - Grad der Selbstständigkeit
  - Qualität des Produktes
  - Reflexion des eigenen Handelns
  - Kooperation mit dem Lehrenden / Aufnahme von Beratung



#### **2.4.1.4 Grundsätze der Leistungsrückmeldung und Beratung**

Die Grundsätze der Leistungsbewertung werden zu Beginn eines jeden Halbjahres den Schüler:innen und Schülern transparent gemacht. Leistungsrückmeldungen können erfolgen

- Nach einer mündlichen Prüfung
- Bei Rückgabe von schriftlichen Leistungsüberprüfungen
- Nach Abschluss eines Projektes
- Nach einem Vortrag oder einer Präsentation
- Bei auffälligen Leistungsveränderungen
- Auf Anfrage
- Als Quartalsfeedback
- Zu Eltern- und Schülersprechtagen

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

#### **2.4.1.5 Bildung der Zeugnisnote**

In die Note gehen alle im Unterricht erbrachten Leistungen ein. Dabei nimmt die Beurteilung der schriftlichen Leistungen den gleichen Stellenwert wie die sonstigen Leistungen ein. Zudem ist bei der Notenfindung die individuelle Lernentwicklung der Schüler\*innen angemessen zu berücksichtigen.