

**Heinrich-Mann-Gymnasium Fühlinger Weg 4  
50765 Köln**



**Schulinterner Lehrplan für das Fach Informatik  
zum Kernlehrplan für die gymnasiale Oberstufe**

Stand: Juni 2019

# Inhaltsverzeichnis

<b>1. Die Fachgruppe Informatik des Heinrich-Mann-Gymnasiums Köln.....</b>	<b>6</b>
<b>2. Entscheidungen zum Unterricht .....</b>	<b>7</b>
2.1. Unterrichtsvorhaben .....	7
2.1.1. Übersichtsraster Unterrichtsvorhaben .....	8
I) Einführungsphase .....	8
II) Qualifikationsphase.....	12
2.1.2. Konkretisierte Unterrichtsvorhaben.....	19
I) Einführungsphase .....	19
II) Qualifikationsphase– Grundkurs/Leistungskurs.....	38
2.2. Grundsätze der fachmethodischen und fachdidaktischen Arbeit .....	69
2.3. Grundsätze der Leistungsbewertung und Leistungsrückmeldung.....	69
2.3.1. Beurteilungsbereich Klausuren .....	70
2.3.2. Beurteilungsbereich Sonstige Mitarbeit .....	71

## 1. Die Fachgruppe Informatik des Heinrich-Mann-Gymnasiums Köln

Beim Heinrich-Mann-Gymnasium handelt es sich um eine in der Regel fünfzügige Schule im Norden von Köln mit zurzeit ca. 1250 Schülerinnen und Schülern sowie ca. 120 Lehrerinnen und Lehrern. Das Einzugsgebiet der Schule umfasst die nördlichen Stadtteile Kölns.

Für den Informatik- und anderen Fachunterricht stehen drei große Computerräume mit je 20 Computerarbeitsplätzen zur Verfügung. Für kleinere Lerngruppen gibt es darüber hinaus drei weitere Räume mit bis zu 15 Computerarbeitsplätzen sowie ein Selbstlernzentrum mit 10 Plätzen. Alle Arbeitsplätze sind an das schulinterne Rechnernetz angeschlossen und alle Schülerinnen und Schüler sowie alle Lehrkräfte verfügen über einen eigenen Zugang in diesem Netzwerk. Jeder Benutzer hat auf dem zentralen Server einen eigenen Speicherbereich und Zugriff auf den Speicherbereich bestimmter Gruppen. Somit haben alle Benutzer von jedem Arbeitsplatz aus Zugriff auf ihre eigenen Daten und auf das Internet.

Der Unterricht erfolgt in Unterrichtsstunden á 67,5 Minuten.

Das Fach Informatik wird am Heinrich-Mann-Gymnasium ab der Jahrgangsstufe 8 im Wahlpflichtbereich II (WP II) zweistündig unterrichtet. In der zweijährigen Laufzeit dieser Kurse wird in altersstufengerechter Weise unter anderem in die Grundlagen der Algorithmik im Bereich der Homepageprogrammierung eingeführt. Zusätzlich wird die Programmierung von Lego Robotern gelehrt.

In der Jahrgangsstufe 5 und 6 findet ein Modellversuch Informatik in der Sekundarstufe I statt.

Im Bereich der Sekundarstufe II werden zurzeit Grund- und Leistungskurse angeboten. Um den Schülerinnen und Schülern gerecht zu werden, die in der Sekundarstufe I keinen Informatikunterricht besucht haben, wird in Kursen der Einführungsphase besonderer Wert daraufgelegt, dass keine Vorkenntnisse aus der Sekundarstufe I zum erfolgreichen Durchlaufen des Kurses erforderlich sind.

Im Unterricht der Sekundarstufe II wird Java als Programmiersprache verwendet. Um weiterhin von Anfang an die Prinzipien der objektorientierten Programmierung berücksichtigen zu können, benutzen wir BlueJ als Entwicklungsumgebung. Für das Erstellen graphischer Programme kann zusätzlich eine didaktische Bibliothek zum Einsatz kommen.

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

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

Die Fachschaft Informatik des Heinrich-Mann-Gymnasiums besteht zurzeit aus vier Lehrkräften mit Oberstufenzulassung.

## 2. Entscheidungen zum Unterricht

### 2.1. Unterrichtsvorhaben

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

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

Im „Übersichtsraster Unterrichtsvorhaben“ wird die für alle Lehrerinnen und Lehrer gemäß Fachkonferenzbeschluss verbindliche Verteilung der Unterrichtsvorhaben dargestellt. Das Übersichtsraster dient dazu, den Kolleginnen und Kollegen einen schnellen Überblick über die Zuordnung der Unterrichtsvorhaben zu den einzelnen Jahrgangsstufen sowie den im Kernlehrplan genannten Kompetenzen, Inhaltsfeldern und inhaltlichen Schwerpunkten zu verschaffen. Der ausgewiesene Zeitbedarf versteht sich als grobe Orientierungsgröße, die nach Bedarf über- oder unterschritten werden kann. Bei dem angegebenen Zeitbedarf handelt es sich um 67,5 Minuten-Stunden.

Um Freiraum für Vertiefungen, besondere Schülerinteressen, aktuelle Themen bzw. die Erfordernisse anderer besonderer Ereignisse (z.B. Praktika, Kursfahrten o.ä.) zu erhalten, wurden im Rahmen dieses schulinternen Lehrplans ca. 75 Prozent der Bruttounterrichtszeit verplant.

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

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

## 2.1.1. Übersichtsraster Unterrichtsvorhaben

### I) Einführungsphase

Einführungsphase	
<p>Unterrichtsvorhaben E-I</p> <p>Thema: Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten</p>	<p>Unterrichtsvorhaben E-II</p> <p>Thema: Grundlagen der objektorientierten Analyse, Modellierung und Implementierung anhand von statischen Grafikszenen</p>
<p><b>Zentrale Kompetenzen:</b>            Argumentieren            Darstellen und Interpretieren            Kommunizieren und Kooperieren</p> <p><b>Inhaltsfelder:</b>            Informatiksysteme            Informatik, Mensch und Gesellschaft</p> <p><b>Inhaltliche Schwerpunkte:</b>            Einzelrechner            Dateisystem            Internet            Einsatz von Informatiksystemen</p> <p><b>Zeitbedarf:</b> 6 Stunden</p>	<p><b>Zentrale Kompetenzen:</b>            Modellieren            Implementieren            Darstellen und Interpretieren            Kommunizieren und Kooperieren</p> <p><b>Inhaltsfelder:</b>            Daten und ihre Strukturierung            Formale Sprachen und Automaten</p> <p><b>Inhaltliche Schwerpunkte:</b>            Objekte und Klassen            Syntax und Semantik einer Programmiersprache</p> <p><b>Zeitbedarf:</b> 7 Stunden</p>

<b>Einführungsphase</b>	
<p>Unterrichtsvorhaben E-IIIa</p> <p>Thema: Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen in Java anhand von einfachen Animationen</p>	<p>Unterrichtsvorhaben E-IIIb</p> <p>Thema: Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen in Java anhand von Variablenoperationen</p>
<p><b>Zentrale Kompetenzen:</b>            Argumentieren            Modellieren            Implementieren            Kommunizieren und Kooperieren</p> <p><b>Inhaltsfelder:</b>            Daten und ihre Strukturierung            Algorithmen            Formale Sprachen und Automaten</p> <p><b>Inhaltliche Schwerpunkte:</b>            Objekte und Klassen            Syntax und Semantik einer Programmiersprache            Analyse, Entwurf und Implementierung einfacher Algorithmen</p> <p><b>Zeitbedarf:</b> 16 Stunden</p>	<p><b>Zentrale Kompetenzen:</b>            Argumentieren            Modellieren            Implementieren            Kommunizieren und Kooperieren</p> <p><b>Inhaltsfelder:</b>            Daten und ihre Strukturierung            Algorithmen            Formale Sprachen und Automaten</p> <p><b>Inhaltliche Schwerpunkte:</b>            Objekte und Klassen            Syntax und Semantik einer Programmiersprache            Analyse, Entwurf und Implementierung einfacher Algorithmen</p> <p><b>Zeitbedarf:</b> 6 Stunden</p>

<b>Einführungsphase</b>	
<p>Unterrichtsvorhaben E-IV</p> <p>Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand von grafischen Spielen und Simulationen</p>	<p>Unterrichtsvorhaben E-V</p> <p>Thema: Such- und Sortieralgorithmen anhand kontextbezogener Beispiele</p>
<p><b>Zentrale Kompetenzen:</b>            Argumentieren            Modellieren            Implementieren            Darstellen und Interpretieren            Kommunizieren und Kooperieren</p> <p><b>Inhaltsfelder:</b>            Daten und ihre Strukturierung            Algorithmen            Formale Sprachen und Automaten</p> <p><b>Inhaltliche Schwerpunkte:</b>            Objekte und Klassen            Syntax und Semantik einer Programmiersprache            Analyse, Entwurf und Implementierung einfacher Algorithmen</p> <p><b>Zeitbedarf:</b> 18 Stunden</p>	<p><b>Zentrale Kompetenzen:</b>            Argumentieren            Modellieren            Darstellen und Interpretieren            Kommunizieren und Kooperieren</p> <p><b>Inhaltsfelder:</b>            Algorithmen</p> <p><b>Inhaltliche Schwerpunkte:</b>            Algorithmen zum Suchen und Sortieren            Analyse, Entwurf und Implementierung einfacher Algorithmen</p> <p><b>Zeitbedarf:</b> 9 Stunden</p>

Einführungsphase	
Unterrichtsvorhaben E-VI  Thema: Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes	
<p><b>Zentrale Kompetenzen:</b>            Argumentieren            Darstellen und Interpretieren            Kommunizieren und Kooperieren</p> <p><b>Inhaltsfelder:</b>            Informatik, Mensch und Gesellschaft            Informatiksysteme</p> <p><b>Inhaltliche Schwerpunkte:</b>            Wirkungen der Automatisierung            Geschichte der automatischen Datenverarbeitung            Digitalisierung</p> <p><b>Zeitbedarf:</b> 13 Stunden</p>	
<b>Summe Einführungsphase: 75</b>	



## II) Qualifikationsphase

Qualifikationsphase I	
Unterrichtsvorhaben Q1-I Thema: Rekursive Algorithmen	
<p>GK</p> <p><b>Zentrale Kompetenzen:</b>            Argumentieren            Modellieren            Implementieren            Darstellen und Interpretieren            Kommunizieren und Kooperieren</p> <p><b>Inhaltsfelder:</b>            Algorithmen            Formale Sprachen und Automaten            Informatiksysteme            Informatik, Mensch und Gesellschaft</p> <p><b>Inhaltliche Schwerpunkte:</b>            Analyse, Entwurf und Implementierung von Algorithmen            Algorithmen in ausgewählten informatischen Kontexten            Syntax und Semantik einer Programmiersprache            Nutzung von Informatiksystemen            Grenzen der Automatisierung</p> <p><b>Zeitbedarf:</b> 15 Stunden</p>	<p>Zusätzlich LK</p> <p>Backtracking, Grenzen der Automatisierbarkeit</p> <p><b>Inhaltliche Schwerpunkte:</b>            Erarbeitung und Implementierung von Backtracking Algorithmen.            Implementierung und Effizienzanalyse von rekursiven Such- und Sortieralgorithmen</p> <p><b>Zeitbedarf:</b> 10 Stunden</p>

## Qualifikationsphase I

Unterrichtsvorhaben Q1-II

Thema: Modellierung und Implementierung dynamischer Listenstrukturen und deren Anwendungen

GK

### Zentrale Kompetenzen:

Argumentieren

Modellieren

Implementieren

Darstellen und Interpretieren

Kommunizieren und Kooperieren

### Inhaltsfelder:

Daten und ihre Strukturierung

Algorithmen

Formale Sprachen und Automaten

Informatiksysteme

### Inhaltliche Schwerpunkte:

Objekte und Klassen

Analyse, Entwurf und Implementierung von Algorithmen

Algorithmen in ausgewählten informatischen Kontexten

Syntax und Semantik einer

Programmiersprache

Nutzung von Informatiksystemen

**Zeitbedarf:** 25 Stunden

Zusätzlich LK

Vertiefung von linearen Datenstrukturen

### Inhaltliche Schwerpunkte:

Aufbau und Implementierung von linearen Datenstrukturen (z.B. Stack, Liste)

Sonderformen von linearen Datenstrukturen (z.B. doppelt verkettete Liste, Ring...)

**Zeitbedarf:** 10 Stunden

## Qualifikationsphase I

Unterrichtsvorhaben Q1-III

Thema: Modellierung und Implementierung dynamische nichtlineare Datenstrukturen am Beispiel der Binärbäume (GK)/ Graphen (LK)

<p>GK</p> <p><b>Zentrale Kompetenzen:</b> Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p><b>Inhaltsfelder:</b> Daten und ihre Strukturierung Algorithmen Formale Sprachen und Automaten Informatiksysteme</p> <p><b>Inhaltliche Schwerpunkte:</b> Objekte und Klassen Analyse, Entwurf und Implementierung von Algorithmen Algorithmen in ausgewählten informatischen Kontexten Syntax und Semantik einer Programmiersprache Nutzung von Informatiksystemen</p> <p><b>Zeitbedarf:</b> 20 Stunden</p>	<p>Zusätzlich LK</p> <p><b>Inhaltliche Schwerpunkte:</b> Objekte und Klassen Analyse, Entwurf und Implementierung von Algorithmen Algorithmen in ausgewählten informatischen Kontexten</p> <p><b>Zeitbedarf:</b> 15 Stunden</p>
---	---

## Qualifikationsphase I

Unterrichtsvorhaben Q1-IV

Thema: Sicherheit und Datenschutz in Netzstrukturen, Modellierung von Client-Serveranwendungen (LK)

<p>GK</p> <p><b>Zentrale Kompetenzen:</b>          Argumentieren          Darstellen und Interpretieren          Kommunizieren und Kooperieren</p> <p><b>Inhaltsfelder:</b>          Informatiksysteme          Informatik, Mensch und Gesellschaft</p> <p><b>Inhaltliche Schwerpunkte:</b>          Einzelrechner und Rechnernetzwerke          Sicherheit          Nutzung von Informatiksystemen,          Wirkungen der Automatisierung</p> <p><b>Zeitbedarf:</b> 15 Stunden</p>	<p>Zusätzlich LK</p> <p><b>Zentrale Kompetenzen:</b>          Argumentieren          Modellieren          Implementieren          Darstellen und Interpretieren          Kommunizieren und Kooperieren</p> <p><b>Inhaltsfelder:</b>          Daten und ihre Strukturierung          Algorithmen          Formale Sprachen und Automaten          Informatiksysteme</p> <p><b>Inhaltliche Schwerpunkte:</b>          Objekte und Klassen          Analyse, Entwurf und Implementierung von Algorithmen          Algorithmen in ausgewählten informatischen Kontexten          Syntax und Semantik einer Programmiersprache          Einzelrechner und Rechnernetzwerke          Nutzung von Informatiksystemen</p> <p><b>Zeitbedarf:</b> 10 Stunden</p>
--	--

**Summe Qualifikationsphase I: 80/ 130**

## Qualifikationsphase II

Unterrichtsvorhaben Q2-I

Thema: Modellierung und Nutzung relationaler Datenbanken in Anwendungskontexten

GK

Zusätzlich LK

**Zentrale Kompetenzen:**

Argumentieren

Modellieren

Implementieren

Darstellen und Interpretieren

Kommunizieren und Kooperieren

Analyse, Entwurf und Implementierung von Algorithmen in ausgewählten Kontexten.

**Zeitbedarf:** 5 Stunden

**Inhaltsfelder:**

Daten und ihre Strukturierung

Algorithmen

Formale Sprache und Automaten

Informatiksysteme

Informatik, Mensch und Gesellschaft

**Inhaltliche Schwerpunkte:**

Datenbanken

Algorithmen in ausgewählten informatischen Kontexten

Syntax und Semantik einer

Programmiersprache

Nutzung von Informatiksystemen

Sicherheit

Wirkung der Automatisierung

**Zeitbedarf:** 25 Stunden

## Qualifikationsphase II

Unterrichtsvorhaben Q2-II

Thema: Technische Informatik

GK

### Zentrale Kompetenzen:

Argumentieren

Modellieren

Implementieren

Darstellen und Interpretieren

Kommunizieren und Kooperieren

### Inhaltsfelder:

Daten und ihre Strukturierung

Algorithmen

Formale Sprachen und Automaten

Informatiksysteme

### Inhaltliche Schwerpunkte:

Objekte und Klassen

Analyse, Entwurf und Implementierung von Algorithmen

Algorithmen in ausgewählten

informatischen Kontexten

Syntax und Semantik einer

Programmiersprache

Nutzung von Informatiksystemen

Zahlwertsystem, Arbeiten mit logischen

Operatoren

**Zeitbedarf:** 5 Stunden

Zusätzlich LK

### Inhaltliche Schwerpunkte:

Objekte und Klassen

Analyse, Entwurf und Implementierung von Algorithmen

Algorithmen in ausgewählten

informatischen Kontexten

Von- Neumann Architektur

Grenzen der Automatisierbarkeit

**Zeitbedarf:** 5 Stunden

<b>Qualifikationsphase II</b>	
Unterrichtsvorhaben Q2-III	
Thema: Endliche Automaten und Formale Sprachen	
<p>GK</p> <p><b>Zentrale Kompetenzen:</b>            Argumentieren            Modellieren            Implementieren            Darstellen und Interpretieren            Kommunizieren und Kooperieren</p> <p><b>Inhaltsfelder:</b>            Formale Sprachen und Automaten            Informatiksysteme</p> <p><b>Inhaltliche Schwerpunkte:</b>            Endliche Automaten            Grammatiken regulärer Sprachen            Möglichkeiten und Grenzen von Automaten            und formalen Sprachen            Nutzung von Informatiksystemen</p> <p><b>Zeitbedarf:</b> 25 Stunden</p>	<p>Zusätzlich LK</p> <p>Kellerautomaten/ kontextfreie Sprachen,            Compiler, Scanner, Parser</p> <p><b>Zentrale Kompetenzen:</b>            Argumentieren            Modellieren            Implementieren            Darstellen und Interpretieren            Kommunizieren und Kooperieren</p> <p><b>Inhaltsfelder:</b>            Daten und ihre Strukturierung            Algorithmen            Formale Sprachen und Automaten            Informatiksysteme</p> <p><b>Inhaltliche Schwerpunkte:</b>            Endliche Automaten und Kellerautomaten            Grammatiken regulärer und kontextfreie            Sprachen            Scanner, Parser und Interpreter für eine            reguläre Sprache            Nutzung von Informatiksystemen</p> <p><b>Zeitbedarf:</b> 15 Stunden</p>
<b>Summe Qualifikationsphase II: 55/ 80</b>	

## 2.1.2. Konkretisierte Unterrichtsvorhaben

Im Folgenden sollen die in Abschnitt 2.1.1. aufgeführten Unterrichtsvorhaben konkretisiert werden. Diese Konkretisierung hat vorschlagenden Charakter, ohne die pädagogische Freiheit des Lehrenden einschränken zu wollen.

### I) Einführungsphase

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

Die Schülerinnen und Schüler

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

#### Unterrichtsvorhaben EF-I

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

Leitfragen: Womit beschäftigt sich die Wissenschaft der Informatik? Wie kann die in der Schule vorhandene informatische Ausstattung genutzt werden?

Absprachen zur vorhabenbezogenen Konkretisierung:

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

Zunächst wird auf den Begriff der Information eingegangen und die Möglichkeit der Kodierung in Form von Daten thematisiert. Anschließend wird auf die Übertragung von Informationen bzw. Daten im Sinne des Sender-Empfänger-Modells eingegangen. Dabei wird eine überblickartige Vorstellung der Kommunikation von Rechnern in Netzwerken erarbeitet.

Des Weiteren soll der grundlegende Aufbau eines Rechnersystems im Sinne der Von-Neumann-Architektur erarbeitet werden und mit dem



grundlegenden Prinzip der Datenverarbeitung (Eingabe-Verarbeitung-Ausgabe) in Beziehung gesetzt werden.

Bei der Beschäftigung mit Datenkodierung, Datenübermittlung und Datenverarbeitung ist jeweils ein Bezug zur konkreten Nutzung der informatischen Ausstattung der Schule herzustellen. So wird in die verantwortungsvolle Nutzung dieser Systeme eingeführt.

Lernmittel / Materialien:

- Arbeitsblätter
- Eine didaktische Entwicklungsumgebung (BlueJ)

Zeitbedarf: ca. 6 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Information, deren Kodierung und Speicherung <ul style="list-style-type: none"> <li>• Informatik als Wissenschaft der Verarbeitung von Informationen</li> <li>• Darstellung von Informationen in Schrift, Bild und Ton</li> <li>• Speichern von Daten mit informatischen Systemen am Beispiel der Schulrechner</li> <li>• Vereinbarung von Richtlinien zur Datenspeicherung auf den Schulrechnern (z.B. Ordnerstruktur, Dateibezeichner usw.)</li> </ul>	Die Schülerinnen und Schüler <ul style="list-style-type: none"> <li>• nutzen die im Unterricht eingesetzten Informatiksysteme selbständig, sicher, zielführend und verantwortungsbewusst (D)</li> </ul>	<ul style="list-style-type: none"> <li>• Kodierung und Dekodierung von Texten mit unbekanntem Zeichensätzen (z.B. Windings)</li> <li>• Kodierung von Bildinformationen in Raster- und Vektorgrafiken</li> </ul>

<p>2. Informations- und Datenübermittlung in Netzen</p> <ul style="list-style-type: none"> <li>• „Sender-Empfänger-Modell“ und seine Bedeutung für die Eindeutigkeit von Kommunikation</li> <li>• Informatische Kommunikation in Rechnernetzen am Beispiel des Schulnetzwerks (z.B. Benutzeranmeldung, Netzwerkordner, Zugriffsrechte, Client-Server.</li> <li>• Grundlagen der technischen Umsetzung von Rechnerkommunikation am Beispiel des Internets (z.B. Netzwerkadresse, Paketvermittlung, Protokoll)</li> <li>• Richtlinien zum verantwortungsvollen Umgang mit dem Internet</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D).</li> <li>• nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K).</li> </ul>	<ul style="list-style-type: none"> <li>• Rollenspiel zur Paketvermittlung im Internet Schülerinnen und Schüler übernehmen die Rollen von Clients und Routern. Sie schicken spielerisch Informationen und Karten von einem Schüler-Client zum anderen. (z.B. Jede Schülerin und jeder Schüler hat eine Adresse, jeder Router darüber hinaus eine Routinetabelle. Mit Hilfe der Tabelle und einem Würfel wird entschieden, wie ein Paket weitervermittelt wird.)</li> </ul>
<p>3. Aufbau informatischer Systeme</p> <ul style="list-style-type: none"> <li>• Identifikation typischer Komponenten informatischer Systeme und anschließende Beschränkung auf das Wesentliche, Herleitung der „Von-Neumann-Architektur“</li> <li>• Identifikation des EVA-Prinzips (Eingabe-Verarbeitung-Ausgabe) als Prinzip der Verarbeitung von Daten und Grundlage der „Von-Neumann-Architektur“</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• beschreiben und erläutern den Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A).</li> </ul>	<ul style="list-style-type: none"> <li>• Durch Demontage eines Demonstrationsrechners entdecken Schülerinnen und Schüler die verschiedenen Hardwarekomponenten eines Informatiksystems. Als Demonstrationsrechner bietet sich ein ausrangierter Schulrechner an. (Eventuell wird auch ein Rechner aus alten Einzelteilen neu zusammengebaut)</li> </ul>

## Unterrichtsvorhaben EF-II

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

Leitfrage: Wie lassen sich Gegenstandsbereiche informatisch modellieren und im Sinne einer Simulation informatisch realisieren?

### Absprachen zur vorhabenbezogenen Konkretisierung:

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

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

Zur graphischen Programmierung können die Beispielklassen Square, Circle und Triangle aus dem Shapes- Projekt oder eine graphische Zusatzbibliothek (basis- Bibliothek) verwendet werden. Die von der Bibliothek vorgegebenen Klassen werden von Schülerinnen und Schülern in Teilen analysiert und entsprechende Objekte anhand einfacher Problemstellungen erprobt. Dazu muss der grundlegende Aufbau einer Java-Klasse thematisiert und zwischen Deklaration, Initialisierung und Methodenaufrufen unterschieden werden.

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

Zeitbedarf: 7 Stunden

## Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Identifikation von Objekten</p> <ul style="list-style-type: none"> <li>• Am Beispiel eines lebensweltnahen Beispiels werden Objekte im Sinne der Objektorientierten Modellierung eingeführt.</li> <li>• Objekte werden mit Objektkarten visualisiert und mit sinnvollen Attributen und „Fähigkeiten“, d.h. Methoden versehen.</li> <li>• Manche Objekte sind prinzipiell typgleich und werden so zu einer Objektsorte bzw. Objektklasse zusammengefasst.</li> <li>• Vertiefung: Modellierung weiterer Beispiele ähnlichen Musters</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li> <li>• modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M),</li> <li>• stellen die Kommunikation zwischen Objekten grafisch dar (M),</li> <li>• implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),</li> <li>• stellen den Zustand eines Objekts dar (D).</li> </ul>	<ul style="list-style-type: none"> <li>• Von Schülerinnen und Schüler werden aus einfachen Figuren Häuser als gleichartige Objekte erstellt, die in einer Klasse mit Attributen und Methoden zusammengefasst werden können.</li> <li>• Aus einfachen Figurenklassen (Dreiecke, Kreise, Quadrate) werden Objekte erstellt, die zur Kölner Skyline zusammengefasst werden können.</li> <li>• Materialien: Ergänzungsmaterialien zum Lehrplannavigator basis- Bibliothek shapes- Projekt (Bestandteil von blueJ)</li> </ul>
<p>2. Analyse von Klassen didaktischer Lernumgebungen</p> <ul style="list-style-type: none"> <li>• Objektorientierte Programmierung als modularisiertes Vorgehen (Entwicklung von Problemlösungen auf Grundlage vorhandener Klassen)</li> <li>• Teilanalyse der Klassen des shapes-Projekt oder der basis- Bibliothek</li> </ul>		<ul style="list-style-type: none"> <li>• Materialien: Dokumentation der didaktischen basis-Bibliothek</li> </ul>
<p>3. Implementierung zweidimensionaler, statischer Szenen</p> <ul style="list-style-type: none"> <li>• Grundaufbau einer Java-Klasse</li> <li>• Konzeption einer Szene</li> <li>• Deklaration und Initialisierung von Objekten</li> </ul> <p>Methodenaufrufe mit Parameterübergabe zur Manipulation von Objekteigenschaften (z.B. Farbe, Position, Größe)</p>		<ul style="list-style-type: none"> <li>• Schülerinnen und Schüler erstellen ein Programm, das mit Hilfe von geometrischen Objekten der basis-Bibliothek bzw. des shapes- Projektes ein Dorf auf den Bildschirm bringt.</li> <li>• Die Schülerinnen und Schüler bilden das Emblem der olympischen Ringe nach.</li> </ul>

## Unterrichtsvorhaben EF-IIIa

Thema: Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen in Java anhand von einfachen Animationen

Leitfragen: Wie lassen sich Animationen und Simulationen optischer Gegenstandsbereiche realisieren?

### Absprachen zur vorhabenbezogenen Konkretisierung:

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

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

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

Zeitbedarf: 16 Stunden

## Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Bewegungsanimationen am Beispiel einfacher grafischer Objekte</p> <ul style="list-style-type: none"> <li>• Kontinuierliche Verschiebung eines Objekts mit Hilfe einer Schleife (While-Schleife)</li> <li>• Bedingungsabfrage zur Realisierung einer Schleifenbedingung für eine Animationsschleife</li> <li>• Mehrstufige Animationen mit mehreren sequenziellen Schleifen</li> <li>• Berechnung von Abständen zwischen Objekten mit Hilfsvariablen</li> <li>• Meldungen zur Kollision zweier Objekte mit Hilfe von Abstandsberechnungen und Verzweigungen (IF-Anweisungen)</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• analysieren und erläutern einfache Algorithmen und Programme (A),</li> <li>• entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M),</li> <li>• ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li> <li>• modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M),</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M),</li> <li>• ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),</li> <li>• modifizieren einfache Algorithmen und Programme (I),</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),</li> <li>• implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I),</li> </ul>	<ul style="list-style-type: none"> <li>• Die Schülerinnen und Schüler realisieren mit Objekten dem shapes- Projekt oder der basis- Bibliothek eine Animation, bei der sich ein Zug über den Bildschirm bewegt. Der Zug soll vor Hindernissen stoppen oder anderen Zügen ausweichen können.</li> </ul>
<p>2. Erstellen und Verwalten größerer Mengen einfacher grafischer Objekte</p> <ul style="list-style-type: none"> <li>• Erzeugung von Objekten mit Hilfe von Zählschleifen (FOR-Schleife)</li> <li>• Verwaltung von Objekten in eindimensionalen Feldern (Arrays)</li> <li>• Animation von Objekten, die in eindimensionalen Feldern (Arrays) verwaltet werden</li> <li>• Vertiefung: Verschiedene Feldbeispiele</li> </ul>	<p>(siehe oben)</p>	<ul style="list-style-type: none"> <li>• Die Schülerinnen und Schüler realisieren eine Reihe von Lampen, die ein- oder ausgeschaltet werden können. die in einem Feld verwaltet werden. Lampen sollen nach bestimmten Regeln an- oder ausgeschaltet werden (Lauflicht).</li> <li>• Die Schülerinnen und Schüler realisieren mit Hilfe mehrerer Quader ein Schachbrett.</li> </ul>

<p>3. Modellierung und Animation komplexerer grafisch repräsentierbarer Objekte</p> <ul style="list-style-type: none"> <li>• Modellierung eines Simulationsprogramms mit eigenen Klassen, die sich selbst mit Hilfe von einfachen Objekten zeigen mit Hilfe eines Implementationsdiagramms</li> <li>• Implementierung eigener Methoden mit und ohne Parameterübergabe</li> <li>• Realisierung von Zustandsvariablen</li> <li>• Thematisierung des Geheimnisprinzips und des Autonomitätsprinzips von Objekten</li> <li>• Animation mit Hilfe des Aufrufs von selbstimplementierten Methoden</li> <li>• Vertiefung: Weitere Projekte</li> </ul>	<ul style="list-style-type: none"> <li>• implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),</li> <li>• testen Programme schrittweise anhand von Beispielen (I),</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I).</li> </ul>	<ul style="list-style-type: none"> <li>• Die Schülerinnen und Schüler modellieren und erstellen eine Klasse, mit deren Hilfe Konfetti simuliert werden kann. Das Konfetti kann an zufälligen Positionen mit zufälliger Farbgebung auf dem Bildschirm erscheinen.</li> <li>• Die Schülerinnen und Schüler erstellen eine Simulation mehrerer Uhren für verschiedene Zeitzonen.</li> <li>• Die Schülerinnen und Schüler erstellen eine Ampelkreuzung mit mehreren Ampelanlagen an einem Bahnübergang</li> </ul>
--	--	---

## Unterrichtsvorhaben EF-IIIb

Thema: Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen in Java anhand von Variablenoperationen

Leitfragen: Wie lassen sich einfache Anwendungen mit Hilfe von Variablenoperationen erstellen?

### Absprachen zur vorhabenbezogenen Konkretisierung:

Der Schwerpunkt dieses Unterrichtsvorhabens liegt auf der Entwicklung mehrerer Projekte, die durch Eingaben des Benutzers und Variablenoperationen einfache Anwendungen erstellen. Zunächst wird ein Projekt bearbeitet, das durch Buchstabenoperationen Strings verändert. Strings sollen z.B. in der Buchstabenfolge verdoppelt, Vokale ersetzt oder umgedreht werden. Für die Umsetzung die des Projekts wird in die unterschiedlichen Variablentypen eingeführt und Stringoperationen erläutert.

Der Umgang mit Zahlvariablen vom Typ integer und double wird im Rahmen eines Taschenrechnerprojektes eingeübt. Das Unterrichtsvorhaben schließt mit einem Projekt, das die Umsetzung eines Taschenrechners zum Gegenstand hat.

Zeitbedarf: 6 Stunden



## Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Veränderung des Inhaltes von String- Variablen</p> <ul style="list-style-type: none"> <li>• Einführung des Datentyps String als zusammengesetzten Datentyp</li> <li>• Umgang mit verschiedenen Datenoperationen des Datentyps String</li> <li>• Bedingungsabfrage bei dem Datentyp String</li> <li>• Neugestaltung von Inhalten in Stringvariablen</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• analysieren und erläutern einfache Algorithmen und Programme (A),</li> <li>• entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M),</li> <li>• ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li> <li>• modellieren (M),</li> <li>• ordnen Attributen, Parametern und</li> <li>• modifizieren einfache Algorithmen und Programme (I),</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),</li> <li>• implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I),</li> <li>• implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),</li> <li>• testen Programme schrittweise anhand von Beispielen (I),</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I).</li> </ul>	<ul style="list-style-type: none"> <li>• Buchstabeninvertierer Der Inhalt einer String- Variable wird umgedreht.</li> <li>• Palindromprüfer. Der Inhalt einer String- Variable wird untersucht, ob er die Bedingung eines Palindroms erfüllt.</li> <li>• 3 Chinesen mit dem Kontrabass. Alle Vokale des Inhaltes einer String- Variable werden durch eine vordefinierte Buchstabenfolge ersetzt.</li> </ul>

<p>2. Umgang mit Inhalten von Zahl- Variablen</p> <ul style="list-style-type: none"><li>• Einführung der verschiedenen Variablentypen von Zahlvariablen</li><li>• Rechnen mit den Inhalten von verschiedenen Zahlvariablen</li></ul>		<ul style="list-style-type: none"><li>• Addierer Der Inhalt zweier Zahlvariablen wird addiert oder subtrahiert.</li><li>• Taschenrechner Der Inhalt zweier Zahlvariablen wird mit Hilfe von einer auswählbaren Rechenoperation verrechnet.</li></ul>
--	--	--

## Unterrichtsvorhaben EF-IV

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

Leitfrage: Wie lassen sich komplexere Datenflüsse und Beziehungen zwischen Objekten und Klassen realisieren?

### Absprachen zur vorhabenbezogenen Konkretisierung:

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

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

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

Zeitbedarf: 17 Stunden

### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Vertiefung des Referenzbegriffs und Einführung des Prinzips der dynamischen Referenzierung</p> <ul style="list-style-type: none"> <li>• Einführung der Objektselektion mit der Maus</li> <li>• Einführung von Objekten zur Auswahl, wie Radiobuttons etc. zur Gestaltung einer graphischen Oberfläche.</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• analysieren und erläutern eine objektorientierte Modellierung (A),</li> <li>• stellen die Kommunikation zwischen Objekten grafisch dar (M),</li> <li>• ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li> <li>• modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M),</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M),</li> <li>• ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),</li> <li>• modellieren Klassen unter Verwendung von Vererbung (M),</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),</li> <li>• modifizieren einfache Algorithmen und Programme (I),</li> <li>• stellen Klassen, Assoziations- und</li> </ul>	<ul style="list-style-type: none"> <li>• Seifenblasen Die Schülerinnen und Schüler entwickeln ein Spiel, bei dem Seifenblasen über den Bildschirm schweben und durch Anklicken mit der Maus zum Zerplatzen gebracht werden können.</li> <li>• Paint Schülerinnen und Schüler entwickeln einen „Nachbau“ des paint- Programmes, bei dem mit der Maus gezeichnet werden kann und Strecken, Formen etc. mit unterschiedlichen Parametern wie Farbe und Dicke konstruiert werden.</li> </ul>

<p>2. Entwicklung eines Spiels mit der Notwendigkeit von Kollisionskontrollen zwischen zwei oder mehr grafischen Objekten</p> <ul style="list-style-type: none"> <li>• Modellierung des Spiels ohne Berücksichtigung der Kollision mit Hilfe eines Implementationsdiagramms</li> <li>• Dokumentation der Klassen des Projekts</li> <li>• Implementierung eines Prototyps ohne Kollision</li> <li>• Ergänzung einer Kollisionsabfrage durch zusätzliche Assoziationsbeziehungen in Diagramm, Dokumentation und Quellcode</li> <li>• Verallgemeinerung der neuen Verwendung von Objektreferenzen</li> <li>• Vertiefung: Entwicklung weiterer Spiele und Simulationen mit vergleichbarer Grundmodellierung</li> </ul>	<p>Vererbungsbeziehungen in Diagrammen grafisch dar (D).</p>	<ul style="list-style-type: none"> <li>• Pongspiel Ein Ball wird mit einem oder später zwei Schiebern auf einer Spielfeldfläche gehalten.</li> <li>• Pong mit Hindernissen Das Pongspiel wird durch Hindernisse, die durch Berührung zum abprallen des Balles oder zum Verschwinden des Hindernisses führen ergänzt.</li> </ul>
--	--	---

<p>3. Erarbeitung einer Simulation mit grafischen Objekten, die sich durch unterschiedliche Ergänzungen voneinander unterscheiden (Vererbung durch Spezialisierung ohne Überschreiben von Methoden)</p> <ul style="list-style-type: none"> <li>• Analyse und Erläuterung einer Basisversion der grafischen Klasse</li> <li>• Realisierung von grafischen Erweiterungen zur Basisklasse mit und ohne Vererbung (Implementationsdiagramm und Quellcode)</li> <li>• Verallgemeinerung und Reflexion des Prinzips der Vererbung am Beispiel der Spezialisierung</li> </ul>	<ul style="list-style-type: none"> <li>• testen Programme schrittweise anhand von Beispielen (I),</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>• dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D).</li> </ul>	<ul style="list-style-type: none"> <li>• Pong mit Hindernissen Die zwei Typen von Hindernissen (abprallen, auslöschen) werden zu einer Oberklasse zusammengefasst.</li> </ul>
<p>4. Entwicklung einer komplexeren Simulation mit grafischen Elementen, die unterschiedliche Animationen durchführen (Vererbung mit Überschreiben von Methoden)</p> <ul style="list-style-type: none"> <li>• Analyse und Erläuterung einer einfachen grafischen Animationsklasse</li> <li>• Spezialisierung der Klasse zu Unterklassen mit verschiedenen Animationen durch Überschreiben der entsprechenden Animationsmethode</li> <li>• Reflexion des Prinzips der späten Bindung</li> <li>• Vertiefung: Entwicklung eines vergleichbaren Projekts mit einer (abstrakten) Oberklasse</li> </ul>		<ul style="list-style-type: none"> <li>• Flummibälle Die Schülerinnen und Schüler entwickeln eine Simulation von Flummibällen, bei der unterschiedliche Bälle unterschiedliche Bewegungen durchführen.</li> <li>• Weihnachtsbaum Die Schülerinnen und Schüler entwickeln eine Simulation eines Weihnachtsbaums mit Hilfe einer abstrakten Klasse Schmuck.</li> <li>• Freies Spiel programmieren Die Schüler und Schülerinnen suchen sich eine einfache Spielidee, die sie dann umsetzen</li> </ul>

## Unterrichtsvorhaben EF-V

Thema: Such- und Sortieralgorithmen anhand kontextbezogener Beispiele

Leitfragen: Wie kann man lineare Datenstrukturen möglichst effizient sortieren bzw. in ihnen suchen?

Absprachen zur vorhabenbezogenen Konkretisierung:

Als Vorbereitung auf die komplexeren Sortieralgorithmen in der Q1 wird das Sortieren auf Arrays mit Hilfe einfacher Algorithmen realisiert. Weiterhin: Sequentielle und binäre Suche in einem Array.

Inhaltliche Schwerpunkte:

Analyse, Entwurf und Implementierung von Minsort, Bubblesort oder auch Insertionsort, sowie die Beurteilung ihrer jeweiligen Effizienz. Bucketsort als konzeptuell anderes Verfahren zum Vergleich (fakultativ).

Zeitbedarf: ca. 10 Stunden

### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Erarbeitung und Darstellung eines Sortieralgorithmus anhand eines praktischen Beispiels	Die Schülerinnen und Schüler <ul style="list-style-type: none"><li>entwickeln einen Sortieralgorithmus (M)</li><li>stellen ihr Ergebnis im Plenum vor (D, K)</li></ul>	Kartenspiel oder nummerierte Klebezettel, Menschengruppe
2. Einführung konkreter Sortierverfahren wie z.B. Minsort, Bubblesort oder Insertionsort. <ul style="list-style-type: none"><li>Erarbeitung der Verfahren in arbeitsteiliger Gruppenarbeit</li><li>Vergleich der Algorithmen bzgl. ihrer Effizienz</li><li>Implementierung</li><li>Überlegungen zur Steigerung der Effizienz der Algorithmen</li><li>Vergleich mit Bucketsort als konzeptuell anderem Verfahren</li></ul>	Die Schülerinnen und Schüler <ul style="list-style-type: none"><li>nutzen die erworbenen informatischen Kenntnisse zur Implementierung (A)</li><li>reflektieren und kommunizieren ihre Ergebnisse im Vergleich der Gruppenergebnisse (K, D)</li></ul>	Zahlenliste mit GUI, Objektarray sortieren



## Unterrichtsvorhaben EF-VI

Thema: Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes

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

### Absprachen zur vorhabenbezogenen Konkretisierung:

Diese Unterrichtseinheit stellt den Abschluss der Einführungsphase dar. Die Schülerinnen und Schüler sollen selbstständig informatische Themenbereiche aus der Geschichte der Datenverarbeitung und insbesondere den daraus sich ergebenden Fragen des Datenschutzes bearbeiten. Diese Themenbereiche werden in Kleingruppen erarbeitet und in Form von Präsentationen (Plakat oder auch elektronisch) vorgestellt. Die Schülerinnen und Schüler sollen dabei – evtl. mit zeitweiser Unterstützung der Lehrkraft – selbstständige Recherchen zu ihren Themen anstellen und auch eine sinnvolle Eingrenzung ihres Themas vornehmen.

Anschließend wird auf den Aspekt des Datenschutzes eingegangen. Dazu wird das Bundesdatenschutzgesetz in Auszügen behandelt und auf schülernahe Beispielsituationen betrachtet. Eine formaljuristische Bewertung der Beispielsituationen kann jedoch im Rahmen des Informatikunterrichts nicht geleistet werden, sondern vielmehr eine persönliche Einschätzung von Fällen im Geiste des Datenschutzgesetzes.

Zeitbedarf: 13 Stunden

## Sequenzierung des Unterrichtsvorhabens:

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

## II) Qualifikationsphase– Grundkurs/Leistungskurs

Die Aufstellung der Unterrichtsvorhaben erfolgt für den Grundkurs und den Leistungskurs kombiniert, da diese zumeist als Huckepack-Kurs durchgeführt werden. Die zusätzlichen Vorhaben für den Leistungskurs werden nach denen des Grundkurses gesondert aufgeführt

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

Die Schülerinnen und Schüler

- verwenden die Fachsprache bei der Kommunikation über informatische Sachverhalte (K),
- nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zur Erschließung, Aufbereitung und Präsentation fachlicher Inhalte (D),
- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung von Dateien unter Berücksichtigung der Rechteverwaltung und Zusammenführung von Arbeitsanteilen (K),
- organisieren und koordinieren kooperatives und eigenverantwortliches Arbeiten (K),
- strukturieren den Arbeitsprozess, vereinbaren Schnittstellen und führen Ergebnisse zusammen (K),
- beurteilen Arbeitsorganisation, Arbeitsabläufe und Ergebnisse (K),
- präsentieren Arbeitsabläufe und -ergebnisse adressatengerecht (K).

Ebenso bieten fast alle Unterrichtsvorhaben, in denen Programme implementiert werden, die Gelegenheit, die folgenden Kompetenzen zu erwerben bzw. zu vertiefen:

Schülerinnen und Schüler

- nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),
- beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),
- interpretieren Fehlermeldungen und korrigieren den Quellcode(I),
- wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I),
- entwickeln mit didaktisch orientierten Entwicklungsumgebungen einfache Benutzungsoberflächen zur Kommunikation mit einem Informatiksystem (M).

## Unterrichtsvorhaben Q1-I

Thema: Rekursive Algorithmen in Anwendungskontexten

Leitfragen: Wie können komplexe, rekursiv definierte Probleme informatisch gelöst werden? Gibt es schnelle (rekursiv definierte) Sortier- und Suchverfahren?

Inhaltsfelder: Inhaltsfeld 2 (Algorithmen), Inhaltsfeld 3 (Formale Sprachen und Automaten), Inhaltsfeld 4 (Informatiksysteme), Inhaltsfeld 5 (Informatik, Mensch und Gesellschaft)

### Absprachen zur vorhabenbezogenen Konkretisierung:

Ausgehend von der Entwicklung effektiver Such- und Sortieralgorithmen wird Rekursion als fundamentale Idee der Informatik im informatischen Zusammenhang angewendet. Dabei wird zwischen linearen und verzweigten Rekursionen unterschieden und das Laufzeitverhalten bei hoher Rekursionstiefe analysiert.

Rekursives Programmieren wird an einzelnen einfachen mathematischen Problemstellungen eingeübt.

Bereits bekannte Such- und Sortierverfahren (z. B. Sortieren durch Einfügen, Sortieren durch Auswahl, Sequentielle Suche) werden rekursiv formuliert und durch leistungsfähigere Verfahren (z. B. Quicksort, Mergesort...) ergänzt.

Der Ansatz von Backtracking kann im Grundkurs thematisiert und ohne Implementierung durchgesprochen werden

### Lernmittel / Materialien:

- Arbeitsblätter
- eine didaktische Entwicklungsumgebung (BlueJ)

Zeitbedarf: ca. 15 Stunden

### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Entwicklung der Rekursion als fundamentale Idee der Informatik</p> <ul style="list-style-type: none"> <li>• rekursive Formeln</li> <li>• rekursive Funktionen / Methoden</li> <li>• rekursive Programmierung</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• analysieren und erläutern Algorithmen und Programme,</li> <li>• modifizieren Algorithmen und Programme,</li> <li>• entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“</li> </ul>	<ul style="list-style-type: none"> <li>• die Vorgehensweise eines rekursiven Suchalgorithmus wird besprochen.</li> <li>• Quicksortvisualisierung zur Erarbeitung der Idee</li> </ul>
<p>2. Rekursion in mathematischen und informatischen Kontexten</p> <ul style="list-style-type: none"> <li>• Rekursion in mathematischen Kontexten</li> <li>• Analyse und Darstellung des rekursiven Ablaufs einer Methode</li> <li>• Analyse des Laufzeitverhaltens linearer und verzweigter Rekursion</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• analysieren und erläutern Algorithmen und Programme,</li> <li>• modifizieren Algorithmen und Programme,</li> <li>• stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar,</li> <li>• testen Programme systematisch anhand von Beispielen und mit Hilfe von Testanwendungen.</li> <li>• untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen.</li> </ul>	<ul style="list-style-type: none"> <li>• Fakultätsfunktion (lineare Rekursion)</li> <li>• Fibonacci-Funktion (verzweigte Rekursion)</li> <li>• ggT (verzweigte Rekursion)</li> <li>• evtl. Fraktale (Kochkurve, Sierpinski dreieck, etc.)</li> </ul>
<p>3. Effiziente Sortierverfahren / Suchverfahren</p> <ul style="list-style-type: none"> <li>• Wiederholung bereits bekannter Sortier- und Suchverfahren als rekursiver Algorithmus</li> <li>• Erarbeitung eines Sortierverfahrens der Laufzeit <math>O(n \cdot \log(n))</math></li> <li>• Erarbeitung eines Suchverfahrens der Laufzeit <math>O(\log(n))</math>.</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• analysieren und erläutern Algorithmen und Programme,</li> <li>• modifizieren Algorithmen und Programme,</li> <li>• stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar,</li> <li>• entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategie „Teilen und Herrschen“,</li> <li>• testen Programme systematisch anhand von Beispielen.</li> <li>• implementieren und erläutern iterative und rekursive Such- und Sortierverfahren</li> </ul>	<ul style="list-style-type: none"> <li>• Demonstrationsprogramm zur Visualisierung von Sortierverfahren</li> </ul>

	<ul style="list-style-type: none"> <li>• beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (<math>A</math>),</li> <li>• untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen.</li> </ul>	
<p>4. evtl. NP-vollständige Probleme lösen mit Backtracking</p> <ul style="list-style-type: none"> <li>• Erarbeitung verschiedener NP-vollständiger Probleme</li> <li>• Algorithmische Beschreibung einer Lösungsidee</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• analysieren und erläutern Algorithmen und Programme,</li> <li>• modifizieren Algorithmen und Programme,</li> <li>• stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar,</li> </ul>	<p>Möglichkeiten</p> <ul style="list-style-type: none"> <li>• Arbeitsblätter und Rasterprogramm zum N-Damenproblem</li> <li>• Arbeitsblätter zum Springerproblem</li> <li>• Rucksackproblem</li> <li>• Irrgartenproblem</li> </ul>

### Absprachen zur vorhabenbezogenen Konkretisierung im Leistungskurs:

Die neuen rekursiven Suchverfahren werden implementiert.

Verschiedene NP-vollständige Probleme (wie z. B. Rucksack, n-Damen, Springer, Irrgarten, etc.) werden algorithmisch rekursiv formuliert und als Backtracking-Algorithmus implementiert.

### Ergänzungen der Unterrichtssequenzen:

Zu 3) Mindestens ein rekursiver Sortieralgorithmus (Mergesort oder Quicksort) wird implementiert.

Untersuchung und Herleitung unterschiedlicher Komplexitätsklassen  
(Speicherbedarf und Laufzeitverhalten),

Zu 4) Implementierung von mindestens einem Problem mit Hilfe des

Backtrackingalgorithmus. Möglichkeiten bei der Implementierung: n-Dameproblem, Rucksackproblem, Türme von Hanoi...  
testen Programme systematisch anhand von Beispielen und mit Hilfe von Testanwendungen.

Untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen.

Zeitbedarf: 10 Stunden

## Unterrichtsvorhaben Q1-II

Thema: Modellierung und Implementierung dynamischer Listenstrukturen und deren Anwendungen

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

Inhaltsfelder: Inhaltsfeld 1 (Daten und ihre Strukturierung), Inhaltsfeld 2 (Algorithmen), Inhaltsfeld 3 (Formale Sprachen und Automaten), Inhaltsfeld 4 (Informatiksysteme)

### Absprachen zur vorhabenbezogene Konkretisierung:

Nach Analyse einer Problemstellung in einem geeigneten Anwendungskontext, in dem Daten nach dem Last-In-First-Out-Prinzip verwaltet werden, werden der Aufbau von Stapeln am Beispiel dargestellt und die Operationen der Klasse Stack anhand der Abiturklasse erläutert. Anschließend werden für die Anwendung notwendige Klassen modelliert und implementiert.

Anschließend wird die Anwendung modifiziert, um den Umgang mit der Datenstruktur zu üben. Anhand einer Anwendung, in der Daten nach dem First-In-First-Out-Prinzip verwaltet werden, werden Unterschiede zwischen den Datenstrukturen Schlange und Stapel erarbeitet. Dabei werden die Operationen der Datenstruktur Schlange thematisiert und die entsprechende Abiturklasse Queue verwendet.

Um einfacher an Objekte zu gelangen, die zwischen anderen gespeichert sind, wird die Klasse List gemäß der Abiturklasse eingeführt und in einem Anwendungskontext verwendet.

(Je nach thematisierten Anwendungskontexten ist eine andere Reihenfolge bei der Behandlung der oben genannten Datenstrukturen möglich und zulässig.)

In mindestens einem weiteren Anwendungskontext wird die Verwaltung von Daten in Schlangen, Stapeln oder Listen vertieft. Modellierungen werden dabei in Entwurfs- und Implementationsdiagrammen dargestellt.

### Lernmittel / Materialien:

- Arbeitsblätter
- eine didaktische Entwicklungsumgebung (BlueJ)

Zeitbedarf: 25 Stunden



### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse Stack</p> <ul style="list-style-type: none"> <li>• Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</li> <li>• Erarbeitung der Funktionalität der Klasse Stack</li> <li>• Modellierung und Implementierung der Anwendung unter Verwendung der Klasse Stack.</li> </ul>	<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,</li> <li>• stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau,</li> <li>• modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten,</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu ,</li> <li>• ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu,</li> <li>• stellen die Kommunikation zwischen Objekten grafisch dar,</li> <li>• stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar,</li> <li>• dokumentieren Klassen,</li> <li>• analysieren und erläutern objektorientierte Modellierungen,</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken.</li> <li>• analysieren und erläutern Algorithmen und Programme,</li> <li>• modifizieren Algorithmen und Programme,</li> <li>• stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar,</li> </ul>	<p>Containerschiff:</p> <ul style="list-style-type: none"> <li>• Einsatz des Algorithmus der Datenstruktur Stack.</li> </ul>
<p>2. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse Queue</p> <ul style="list-style-type: none"> <li>• Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</li> <li>• Erarbeitung der Funktionalität der Klasse Queue</li> <li>• Modellierung und Implementierung der Anwendung unter Verwendung der Klasse Queue.</li> </ul>		<p>Wartezimmer:</p> <ul style="list-style-type: none"> <li>• Einsatz der Datenstruktur Queue.</li> </ul>
<p>3. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse List</p> <ul style="list-style-type: none"> <li>• Erarbeitung der Vorteile der Klasse List im Gegensatz zu den bereits bekannten linearen Strukturen</li> <li>• Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse List.</li> </ul>		<p>Schulverwaltungsprogramm</p> <ul style="list-style-type: none"> <li>• Einsatz der Datenstruktur List.</li> </ul>

	<ul style="list-style-type: none"> <li>• implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen,</li> <li>• testen Programme systematisch anhand von Beispielen und mit Hilfe von Testanwendungen.</li> </ul>	
4. Vertiefung / Anwendung einer linearen Datenstruktur im Anwendungskontext.	<p>zusätzlich: Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren,</li> <li>• verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie</li> </ul>	<p>Umsetzung in einem größeren Projekt, z.B.:</p> <ul style="list-style-type: none"> <li>• Objektorientiertes Spiel Schatzsuche, DalliDalli...</li> <li>• Bibliotheksverwaltung</li> </ul>

Absprachen zur vorhabenbezogenen Konkretisierung im Leistungskurs:

Stack und Queue werden implementiert. Erweiterungen der List- Klasse werden entwickelt.

Eine erweiterte Anwendung im Bereich der linearen Datenstrukturen wird entwickelt und in Teilen implementiert.

Ergänzungen der Unterrichtssequenzen:

Zu 1) Implementierung der Klasse Stack.

Zu 2) Implementierung der Klasse Queue.

Zu 3) Analysiert den Quelltext einzelner Methoden der list- Klasse und rekonstruiert den Aufbau. Graphische Darstellung und Implementierung Operationen erweiterter dynamischer Datenstrukturen wie Ring und doppelt verketteter Liste.

Zeitbedarf: 10 Stunden

## Unterrichtsvorhaben Q1-III

Thema: Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen am Beispiel der Binärbäume// Graphen (im LK)

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

Inhaltsfelder: Inhaltsfeld 1 (Daten und ihre Strukturierung), Inhaltsfeld 2 (Algorithmen), Inhaltsfeld 3 (Formale Sprachen und Automaten), Inhaltsfeld 4 (Informatiksysteme)

### Absprachen zur vorhabenbezogenen Konkretisierung:

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

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

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

Zu einer Problemstellung in einem entsprechenden Anwendungskontext werden die Operationen der Datenstruktur Suchbaum thematisiert und unter der Verwendung der Klasse BinarySearchTree der Vorgaben für das Zentralabitur weitere Klassen oder Methoden in diesem Kontext modelliert und implementiert. Die Suchbäume werden wie zuvor auch grafisch dargestellt.

Die Verwendung von binären Bäumen und Suchbäumen wird anhand weiterer Problemstellungen oder anderer Kontexte weiter geübt.

### Lernmittel / Materialien:

- eine didaktische Entwicklungsumgebung (BlueJ)
- Arbeitsblätter

Zeitbedarf:20 Stunden

### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Analyse von Baumstrukturen in verschiedenen Kontexten</p> <ul style="list-style-type: none"> <li>• Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit)</li> <li>• Aufbau und Darstellung von binären Bäumen anhand von Baumstrukturen in verschiedenen Kontexten</li> </ul>	<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,</li> <li>• stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau,</li> <li>• modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten,</li> </ul>	<ul style="list-style-type: none"> <li>• Demoprogramm und Arbeitsblätter zum Projekt Tiere-Raten</li> </ul>
<p>2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse BinaryTree</p> <ul style="list-style-type: none"> <li>• Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext</li> <li>• Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms</li> <li>• Erarbeitung der Klasse BinaryTree und beispielhafte Anwendung der Operationen</li> <li>• Implementierung der Anwendung oder von Teilen der Anwendung</li> <li>• Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf</li> </ul>	<ul style="list-style-type: none"> <li>• modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren,</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen lineare und nichtlineare Datensammlungen zu,</li> <li>• verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie,</li> <li>• ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu,</li> <li>• stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar,</li> </ul>	<ul style="list-style-type: none"> <li>• Arbeitsblätter zur Projektarbeit Tiere-Raten</li> <li>• Arbeitsblätter zu Traversierungsverfahren</li> <li>• Arbeitsblätter zur Projektarbeit Morsebaum</li> </ul>
<p>3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse BinarySearchTree</p> <ul style="list-style-type: none"> <li>• Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</li> <li>• Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramm,</li> <li>• grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften</li> </ul>	<ul style="list-style-type: none"> <li>• dokumentieren Klassen,</li> <li>• analysieren und erläutern objektorientierte Modellierungen,</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken.</li> <li>• analysieren und erläutern Algorithmen und Programme,</li> </ul>	<ul style="list-style-type: none"> <li>• Arbeitsblätter zum binären Suchbaum</li> <li>• Arbeitsblätter zum Baumsortieren und zu Traversierungsverfahren</li> </ul>

<ul style="list-style-type: none"> <li>• Erarbeitung der Klasse BinarySearchTree und Einführung des Interface Item zur Realisierung einer geeigneten Ordnungsrelation</li> <li>• Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums</li> </ul>	<ul style="list-style-type: none"> <li>• modifizieren Algorithmen und Programme,</li> <li>• stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar,</li> <li>• entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“,</li> <li>• implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen,</li> <li>• testen Programme systematisch anhand von Beispielen und mit Hilfe von Testanwendungen.</li> <li>• erläutern Operationen dynamischer (linearer nicht-linearer) Datenstrukturen (A),</li> </ul>	
<p>4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen</p>		<ul style="list-style-type: none"> <li>• Projektarbeiten zu einem der Themen Termbäume, Mobilebäume, Morsebäume, Stichwortbaum, Ahnenbaum</li> </ul>

Abspraken zur vorhabenbezogenen Konkretisierung im Leistungskurs:

BinaryTree und ausgewählte Methoden von BinarySearchTree werden implementiert

Kennenlernen des Konzeptes von teilausgeglichenen AVL- Bäumen.

Ergänzungen der Unterrichtssequenzen:

Zu 2) Implementierung der Klasse BinaryTree.

Zu 3) Implementierung ausgewählter Methoden der Klasse BinarySearchTree.

Zu 3) AVL- Bäume algorithmisch erproben.

Thema: Modellierung und Implementierung dynamischer nichtlinearer Datenstrukturen am Beispiel der Graphen

Leitfragen: Bei welchen Problemstellungen reichen die bekannten Datenstrukturen nicht aus? Welche Möglichkeiten gibt es, flexibel miteinander verknüpfte Informationen zu verwalten? Wie hängen die Datenstrukturen Graph, Baum und Liste zusammen?

Inhaltsfelder: Inhaltsfeld 1 (Daten und ihre Strukturierung), Inhaltsfeld 2 (Algorithmen), Inhaltsfeld 3 (Formale Sprachen und Automaten), Inhaltsfeld 4 (Informatiksysteme)

Abspraken zur vorhabenbezogene Konkretisierung:

Nach Analyse einer Problemstellung in einem geeigneten Anwendungskontext (z. B. das Eulerkreisproblem), in dem Daten in Form eines Graphen verwaltet werden, werden der Aufbau und die Darstellungsformen von Graphen am Beispiel dargestellt und ausgewählte Problemstellungen exemplarisch analysiert.

Die Operationen der Klasse Graph werden erläutert und im Anwendungszusammenhang bei der Lösung grundlegender Probleme (wie z.B. der Traversierung eines Graphen) genutzt.

Lernmittel / Materialien:

- Arbeitsblätter auf der Schulhomepage
- eine didaktische Entwicklungsumgebung (BlueJ)

Zeitbedarf: 15 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

<b>Unterrichtssequenzen</b>	<b>zu entwickelnde Kompetenzen</b>	<b>Beispiele, Medien, Materialien</b>
1. Die Datenstruktur des Graphen im Anwendungskontext <ul style="list-style-type: none"> <li>• Grundbegriffe (Knoten, Kanten und anderes)</li> <li>• Darstellungsformen eines Graphen</li> </ul>	Die Schülerinnen und Schüler <ul style="list-style-type: none"> <li>• ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen,</li> <li>• stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau,</li> </ul>	Eulerkreisproblem: <ul style="list-style-type: none"> <li>• Arbeitsblätter zum Eulerweg / Eulerkreis</li> <li>• Visualisierungsprogramm für Graphen</li> </ul>
2. Die Datenstruktur des Graphen im Anwendungskontext unter Nutzung der Klasse Graph <ul style="list-style-type: none"> <li>• Modellierung von Anwendungssituationen als Graph</li> <li>• Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</li> <li>• Erarbeitung der Funktionalität der Klasse Graph</li> <li>• Modellierung und Implementierung verschiedener Problemstellungen unter Verwendung der Klasse Graph.</li> </ul>	<ul style="list-style-type: none"> <li>• modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten,</li> <li>• modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren,</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu,</li> </ul>	Projektarbeit in einem größeren Kontext: <ul style="list-style-type: none"> <li>• Navigationssystem</li> <li>• Weiterführung des Eulerkreisproblems</li> </ul> Arbeitsblätter: <ul style="list-style-type: none"> <li>• Tiefensuche / Breitensuche</li> <li>• kürzeste Wege mit Dijkstra</li> </ul> mögliche Vertiefungen: <ul style="list-style-type: none"> <li>• TravelingSalesman Problem</li> <li>• minimale Spannbäume</li> </ul>

	<ul style="list-style-type: none"><li>• verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie,</li><li>• ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu,</li><li>• stellen die Kommunikation zwischen Objekten grafisch dar,</li><li>• stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar,</li><li>• dokumentieren Klassen,</li><li>• analysieren und erläutern objektorientierte Modellierungen,</li><li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken.</li><li>• analysieren und erläutern Algorithmen und Programme,</li><li>• modifizieren Algorithmen und Programme,</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“,</li><li>• implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen,</li><li>• testen Programme systematisch anhand von Beispielen.</li><li>• erläutern Operationen dynamischer Datenstrukturen,</li></ul>	
--	--	--

## Unterrichtsvorhaben Q1-IV

Thema: Sicherheit und Datenschutz in Netzstrukturen

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

Inhaltsfelder: Inhaltsfeld 2 (Algorithmen), Inhaltsfeld 4 (Informatiksysteme), Inhaltsfeld 5 (Informatik, Mensch und Gesellschaft)

### Absprachen zur vorhabenbezogenen Konkretisierung:

Ausgehend von einer Kommunikation zwischen zwei Kommunikationspartnern über eine einfache Leitung wird die Notwendigkeiten einer Datenübertragung erarbeitet. Die Schichten des TCP/IP-Schichtenmodells werden beispielgebunden erarbeitet (Basisbandübertragungsverfahren, Prüfverfahren, Vermittlungsschicht, Anwendungsprotokoll) und an einer Simulationssoftware getestet. Verschiedene Netzwerk-Topologien werden entwickelt und in Client-Server-Anwendungen simuliert.

Über die Sicherheit von Netzwerkanwendungen wird das Augenmerk auf verschiedene symmetrische und asymmetrische kryptografische Verfahren gelenkt, welche analysiert und erläutert werden. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht runden das Unterrichtsvorhaben ab.

### Lernmittel / Materialien:

- Arbeitsblätter zur Einführung in Netzwerke
- Arbeitsblätter zu kryptografischen Verfahren
- eine didaktische Entwicklungsumgebung (BlueJ)

Zeitbedarf: 15 Stunden



### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Schichten des TCP/IP-Protokolls</p> <ul style="list-style-type: none"> <li>• Erarbeitung der Notwendigkeiten einer Netzwerkkommunikation</li> <li>• Erarbeitung der Schichten des TCP/IP-Protokolls: Basisbandübertragung, Prüfverfahren, Routing/Vermittlungsschicht, Anwendungsprotokolle</li> </ul>	<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,</li> </ul>	<ul style="list-style-type: none"> <li>• Arbeitsblätter zur Einführung in Netzwerke</li> </ul>
<p>2. Simulation von Netzwerken / Netzwerk-Topologien</p> <ul style="list-style-type: none"> <li>• Erarbeitung der Topologien: Peer-to-Peer, Sterntopologie, Baumtopologie, Vermaschtes Netz</li> <li>• Simulation von Client-Server-Anwendungen</li> <li>• Simulation von Protokollen der Anwendungsschicht (POP3, SMTP, etc.)</li> </ul>	<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,</li> </ul>	<ul style="list-style-type: none"> <li>• Arbeitsblätter zu zum POP3 Protokoll</li> </ul>
<p>3. Analyse und Erläuterung kryptografischer Verfahren</p> <ul style="list-style-type: none"> <li>• Erläuterung symmetrischer Verfahren: Cäsar, Vigenère</li> <li>• Erläuterung asymmetrischer Verfahren: RSA</li> <li>• Analyse der Sicherheit verschiedener Verfahren und Auswirkungen auf den Datenschutz/Urheberrecht</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• analysieren und erläutern Algorithmen und Programme,</li> <li>• stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar,</li> <li>• analysieren und erläutern Eigenschaften, Funktionsweisen und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren.</li> <li>• untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts,</li> <li>• untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben,</li> </ul>	<ul style="list-style-type: none"> <li>• Arbeitsblätter zu kryptografischen Verfahren</li> <li>• Materialien von klicksafe (Zusatzmodule „Nicht alles was geht, ist auch erlaubt“, „Ich bin öffentlich ganz privat“, „Datenschutz im Internet“)</li> </ul>

	hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen.	
--	---	--

Absprachen zur vorhabenbezogenen Konkretisierung im Leistungskurs:

Thema: Modellierung und Implementierung von Client-Server-Anwendungen

Inhaltsfelder: Inhaltsfeld 1 (Daten und ihre Strukturierung), Inhaltsfeld 2 (Algorithmen), Inhaltsfeld 3 (Formale Sprache und Automaten), Inhaltsfeld 4 (Informatiksysteme)

Absprachen zur vorhabenbezogenen Konkretisierung:

Ausgehend von einer einfachen Echo-Anwendung werden die beteiligten Komponenten (Echo-Server und Echo-Client) entwickelt und unter Verwendung der ZA-Klassen implementiert.

Die Echo-Anwendung wird zu einer Chat-Anwendung erweitert, notwendige Protokolle werden entwickelt und systematisch dargestellt.

Die Schülerinnen und Schüler entwickeln eine individuelle Client-Server-Anwendung, definieren notwendige Protokolle und erweitern die Chat-Anwendung entsprechend der Vorgaben.

Die Schülerinnen und Schüler entwickeln eine netzwerk-basierte Spielanwendung

Lernmittel / Materialien:

- Arbeitsblätter.
- eine didaktische Entwicklungsumgebung (BlueJ)

Zeitbedarf: 10 Stunden

### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Entwicklung einer Echo-Anwendung <ul style="list-style-type: none"> <li>• Grundlegende Begriffe</li> <li>• ZA-Klassen</li> </ul>	Die Schülerinnen und Schüler <ul style="list-style-type: none"> <li>• ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen,</li> <li>• modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten,</li> <li>• modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren,</li> <li>• verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie,</li> <li>• ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu,</li> <li>• stellen die Kommunikation zwischen Objekten grafisch dar,</li> <li>• stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar,</li> <li>• dokumentieren Klassen,</li> <li>• analysieren und erläutern objektorientierte Modellierungen,</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken.</li> <li>• modifizieren Algorithmen und Programme,</li> <li>• implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen,</li> </ul>	
2. Entwicklung einer Chat-Anwendung <ul style="list-style-type: none"> <li>• Nutzung der Datenstruktur List</li> </ul>		
3. Entwicklung einer beliebigen Client-Server-Anwendung		Umsetzung eines größeren Projektes <ul style="list-style-type: none"> <li>• Netzwerkspiel (wie z.B. Tic TacToe, Schiffeversenken...)</li> <li>• Anwendungsserver (wie z.B. Buchungsserver für Kinokarten)</li> </ul>

	<ul style="list-style-type: none"><li>• testen Programme systematisch anhand von Beispielen.</li><li>• erläutern das Prinzip der Nebenläufigkeit,</li><li>• analysieren und erläutern Algorithmen und Methoden zur Client-Server-Kommunikation,</li><li>• entwickeln und implementieren Algorithmen und Methoden zur Client-Server-Kommunikation.</li><li>• beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken,</li><li>• analysieren und erläutern Protokolle zur Kommunikation in einem Client-Server-Netzwerk,</li><li>• entwickeln und erweitern Protokolle zur Kommunikation in einem Client-Server-Netzwerk.</li></ul>	
--	--	--

## Unterrichtsvorhaben Q2-I

Thema: Modellierung und Nutzung relationaler Datenbanken in Anwendungskontexten

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

### Inhaltliche Schwerpunkte:

- Datenbanken
  - Modellierung von Datenbanken
- Algorithmen in ausgewählten informatischen Kontexten
  - Datenbankabfragen
- Syntax und Semantik einer Programmiersprache
  - Datenbankabfragesprache SQL
- Nutzung von Informatiksystemen
- Wirkung der Automatisierung
  - Sicherheit

Inhaltsfelder: Inhaltsfeld 1 (Daten und ihre Strukturierung), Inhaltsfeld 2 (Algorithmen), Inhaltsfeld 3 (Formale Sprachen und Automaten), Inhaltsfeld 4 (Informatiksysteme), Inhaltsfeld 5 (Informatik, Mensch und Gesellschaft)

### Absprachen zur vorhabenbezogenen Konkretisierung:

Ausgehend von einer konkreten Anwendungssituation entwickeln die Schülerinnen und Schüler Ideen zur Modellierung von Daten und erkennen die Vorzüge von Datenbanksystemen.

In weiteren Anwendungskontexten müssen Datenbanken entwickelt werden, um Daten zu speichern und Informationen für die Beantwortung von möglicherweise auftretenden Fragen zur Verfügung zu stellen. Dafür ermitteln Schülerinnen und Schüler in den Anwendungssituationen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten und stellen diese in Entity-Relationship-Modellen dar. Entity-Relationship-Modelle werden interpretiert und erläutert, modifiziert und in das Relationale Modell überführt.

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

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

Mit Hilfe von SQL-Anweisungen können anschließend im Kontext relevante Informationen aus der Datenbank extrahiert werden. Die Operationen der Relationenalgebra werden mit SQL-Abfragen simuliert.

Anhand von Fallbeispielen werden Probleme bei der Nutzung von Datenbanksystemen aufgezeigt und im Hinblick auf gesellschaftliche Auswirkungen diskutiert.

(Die exakte Abfolge der Teilvorhaben sowie die Verzahnung theoretischer und praktischer Anteile kann hierbei je nach Anwendungskontext flexibel angepasst werden.)

#### Lernmittel / Materialien:

- Arbeitsblätter
- Datenbankserver (z. B. XAMPP)
- SQL-Tutorial (z. B. die Seite SQL-Zoo)

Zeitbedarf: 25 Stunden

## Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Modellierung von relationalen Datenbanken</p> <ul style="list-style-type: none"> <li>• Entity-Relationship-Diagramm: Ermittlung von Entitäten, zugehörigen Attributen, Relationen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms</li> <li>• Erläuterung und Modifizierung einer Datenbankmodellierung</li> <li>• Entwicklung einer Datenbank aus einem Datenbankentwurf: Modellierung eines relationalen Datenbankschemas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln</li> <li>• Redundanz, Konsistenz und Normalformen: Untersuchung einer Datenbank hinsichtlich Konsistenz und Redundanz in einer Anwendungssituation. Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten)</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten,</li> <li>• stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten mit Kardinalitäten in einem Entity-Relationship-Diagramm grafisch dar,</li> <li>• modifizieren eine Datenbankmodellierung,</li> <li>• modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema,</li> <li>• bestimmen Primär- und Sekundärschlüssel,</li> <li>• analysieren und erläutern eine Datenbankmodellierung,</li> <li>• erläutern die Eigenschaften normalisierter Datenbankschemata,</li> <li>• überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften,</li> <li>• überführen Datenbankschemata in die 1. bis 3. Normalform.</li> </ul>	<ul style="list-style-type: none"> <li>• Arbeitsblätter zur Einführung in Datenbanken auf der Schulhomepage: <ul style="list-style-type: none"> <li>○ Modellierung Zeugnis</li> <li>○ Einführung ERM</li> <li>○ ERD Bücherei</li> <li>○ ERM -&gt; RM</li> <li>○ (Relationenalgebra)</li> <li>○ Normalisierung</li> </ul> </li> </ul>
<p>2. Nutzung von relationalen Datenbanken</p> <ul style="list-style-type: none"> <li>• Aufbau von Datenbanken und Grundbegriffe: Entwicklung von Fragestellungen zur vorhandenen Datenbank</li> <li>• Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, Datenbankschema</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• modifizieren eine Datenbankmodellierung,</li> <li>• bestimmen Primär- und Sekundärschlüssel,</li> <li>• analysieren und erläutern eine Datenbankmodellierung,</li> <li>• ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpften Tabellen,</li> </ul>	<ul style="list-style-type: none"> <li>• Arbeitsblätter zur Einführung in SQL</li> <li>• Tutorial der Seite SQL-Zoo (<a href="http://sqlzoo.net/wiki/Main_Page">http://sqlzoo.net/wiki/Main_Page</a>)</li> </ul>

<ul style="list-style-type: none"> <li>• SQL-Abfragen: Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL (SELECT (DISTINCT) ...FROM, WHERE, AND, OR, NOT) auf einer Tabelle Analyse und Erarbeitung von SQL-Abfragen auf einer und mehrerer Tabelle zur Beantwortung der Fragestellungen (JOIN, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, &lt;, &gt;, &lt;=, &gt;=, LIKE, BETWEEN, IN, IS NULL)</li> <li>• Vertiefung an einem weiteren Datenbankbeispiel: Vertiefungen am Beispiel der Relationenalgebra</li> </ul>	<ul style="list-style-type: none"> <li>• analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage,</li> <li>• verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren.</li> <li>• erläutern Eigenschaften und Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung,</li> </ul>	
<p>3. Gesellschaftliche Auswirkungen der Nutzung von Datenbanksystemen</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,</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.</li> </ul>	<ul style="list-style-type: none"> <li>• Fallbeispiele zur Nutzung von Datenbanksystemen</li> <li>• Spiel zum Missbrauch von Daten: DataDealer (<a href="http://datadealer.com/de">http://datadealer.com/de</a>)</li> </ul>

Absprachen zur vorhabenbezogenen Konkretisierung im Leistungskurs:

Zusätzlich kann eine eigene Datenbank konkret aufgestellt und mit Daten versorgt werden.



## Unterrichtsvorhaben Q2-II

Thema: Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

Leitfragen: Was sind die logischen Grundstrukturen eines Computers? Welches sind die grundlegenden Konzepte und Elemente des Computers?

Inhaltsfelder: Inhaltsfeld 4 (Informatiksysteme), Inhaltsfeld 5 (Informatik, Mensch und Gesellschaft)

### Abspraken zur vorhabenbezogenen Konkretisierung:

Anhand von den logischen Grundschaltungen wird die prinzipielle Arbeitsweise von Computern verdeutlicht. Mit der Vorstellung verschiedener Zahlwertsysteme und deren Umrechnung wird Einblick in die inneren Operationen des Computers gegeben

### Lernmittel / Materialien:

Arbeitsblätter

LogicSim zum Aufbau logischer Schaltungen

Zeitbedarf: 5 Stunden

### **Sequenzierung des Unterrichtsvorhabens:**

<b>Unterrichtssequenzen</b>	<b>zu entwickelnde Kompetenzen</b>	<b>Beispiele, Medien, Materialien</b>
1. Einführung in die prädikative Logik: <ul style="list-style-type: none"><li>• Entwicklung eines logischen Ausdrucks aus einer Problemstellung und Erstellung von Wahrheitstafeln</li><li>• Umsetzung der logischen Ausdrücke in eine logische Schaltung</li><li>• Entwicklung einfacher Rechenkomponenten des Computers mit Hilfe von logischen Gattern</li></ul>	Die Schülerinnen und Schüler <ul style="list-style-type: none"><li>• lernen die grundlegenden Gesetze prädikativer Logik kennen und erkenne ihre Bedeutung im inneren Aufbau Computerkomponenten,</li><li>• entwickeln die Fähigkeit mit unterschiedlichen Zahlwertsystemen zu operieren</li></ul>	Arbeitsblätter LogicSim Halb- Volladdierer Codierer/ Decodierer Speicherelemente
2. verschiedene Zahlwertsysteme <ul style="list-style-type: none"><li>• Vorstellung der Zahlwertsysteme</li><li>• Umrechnung der Zahlwertsysteme</li><li>• Rechnen mit Binärzahlen</li></ul>		Arbeitsblätter

Absprachen zur vorhabenbezogenen Konkretisierung im Leistungskurs:

Thema: Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

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

Inhaltsfelder: Inhaltsfeld 4 (Informatiksysteme), Inhaltsfeld 5 (Informatik, Mensch und Gesellschaft)

Absprachen zur vorhabenbezogenen Konkretisierung:

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

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

Lernmittel / Materialien:

- Arbeitsblätter

Zeitbedarf: 5 Stunden

### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Von-Neumann-Architektur und die Ausführung maschinennaher Programme:</p> <ul style="list-style-type: none"> <li>• prinzipieller Aufbau einer von Neumann-Architektur mit CPU, Rechenwerk, Steuerwerk, Register und Hauptspeicher</li> <li>• einige maschinennahe Befehlen und ihre Repräsentation in einem Binär-Code, der in einem Register gespeichert werden kann</li> <li>• Analyse und Erläuterung der Funktionsweise eines einfachen maschinennahen Programms</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• erläutern die Ausführung eines einfachen maschinennahen Programms sowie die Datenspeicherung auf einer „Von-Neumann-Architektur“,</li> <li>• untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen.</li> </ul>	<p>geeigneter Modellrechner</p>
<p>2. Grenzen der Automatisierbarkeit:</p> <ul style="list-style-type: none"> <li>• Vorstellung des Halteproblems</li> <li>• Unlösbarkeit des Halteproblems</li> <li>• Beurteilung des Einsatzes von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen</li> </ul>		<p>Arbeitsblätter von der Schulhomepage zum Halteproblem Mögliche Verknüpfung zu Unterrichtsvorhaben Q1-1 (Backtracking)</p>

## Unterrichtsvorhaben O2-II

Thema: Endliche Automaten und Formale Sprachen

Leitfragen: Wie kann man endliche Automaten genau beschreiben? Wie können endliche Automaten modelliert werden? Wie können Sprachen durch Grammatiken beschrieben werden? Welche Zusammenhänge gibt es zwischen formalen Sprachen, Automaten und Grammatiken?

Inhaltsfelder: Inhaltsfeld 1 (Daten und ihre Strukturierung), Inhaltsfeld 2 (Algorithmen), Inhaltsfeld 3 (Formale Sprachen und Automaten), Inhaltsfeld 4 (Informatiksysteme), Inhaltsfeld 5 (Informatik, Mensch und Gesellschaft)

Abspraken zur vorhabenbezogenen Konkretisierung:

Ausgehend von einem konkreten Anwendungsbeispiel entwickeln die Schülerinnen und Schüler das Modell der Grammatik einer formalen Sprache und das Modell des endlichen Automaten. Die Schülerinnen und Schüler überführen Automaten in verschiedene Darstellungsformen und ermitteln die akzeptierte Sprache eines Automaten (z. B. in Form von regulären Ausdrücken). An einem Beispiel wird ein nichtdeterministischer Akzeptor als Alternative gegenüber einem entsprechenden deterministischen Akzeptor eingeführt.

Der Zusammenhang zwischen endlichen Automaten und regulären Grammatiken wird durch die Entwicklung allgemeingültiger Verfahren zur Transformation zwischen Automaten und Grammatik dargestellt. Die Unzulänglichkeit endlicher Automaten und regulärer Grammatiken wird an Beispielen verdeutlicht.

Lernmittel / Materialien:

- Arbeitsblätter

Zeitbedarf: 25 Stunden

### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Einführung in Automaten/Grammatiken</p> <ul style="list-style-type: none"> <li>• Grammatiken: Grammatik einer natürlichen Sprache Grammatik einer künstlichen Sprache Idee des Parsens</li> <li>• Automaten: erkennender Automat zu Symbolen einer Sprache Modell des endlichen Automaten Darstellungsformen Sprache eines Automaten als regulärer Ausdruck nichtdeterministische Automaten</li> </ul>	<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,</li> <li>• ermitteln die Sprache, die ein endlicher Automat akzeptiert,</li> <li>• entwickeln und modifizieren zu einer Problemstellung endliche Automaten,</li> <li>• stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform,</li> <li>• entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten.</li> </ul>	<p>Mögliche Einstiege:</p> <ul style="list-style-type: none"> <li>• Grundidee eines Compilers: Scanner, Parser, Codierer</li> <li>• Entwicklung eines Termprüfers, Termerkenners.</li> </ul>
<p>2. Zusammenhang zwischen endlichen Automaten und regulären Grammatiken</p> <ul style="list-style-type: none"> <li>• reguläre Grammatik: Definition Anwendungen</li> <li>• Zusammenhang zu endlichen Automaten</li> <li>• Grenzen der endlichen Automaten/regulären Grammatiken</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• analysieren und erläutern Grammatiken regulärer Sprachen,</li> <li>• modifizieren Grammatiken regulärer Sprachen,</li> <li>• ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird,</li> <li>• entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt,</li> <li>• entwickeln zur akzeptierten Sprache eines Automaten eine zugehörige Grammatik,</li> <li>• beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken.</li> </ul>	<p>Arbeitsblätter Orientierungsrahmen: Chomsky</p>

### Absprachen zur vorhabenbezogenen Konkretisierung im Leistungskurs:

Beschreibung und Modellierung eines Kellerautomaten  
Einführung einer kontextfreien Sprache

Das Grammatikmodell der regulären Grammatiken wird auf das Modell der kontextfreien Grammatiken erweitert und die Auswirkungen auf das entsprechende Automatenmodell der Kellerautomaten veranschaulicht. Die Unzulänglichkeit der Kellerautomaten und kontextfreien Grammatiken wird an Beispielen verdeutlicht.

Eine Implementierung eines Kellerautomaten wird besprochen und durchgeführt.

Zeitbedarf: 5 Stunden

### **Sequenzierung des Unterrichtsvorhabens:**

<b>Unterrichtssequenzen</b>	<b>zu entwickelnde Kompetenzen</b>	<b>Beispiele, Medien, Materialien</b>
3. Zusammenhang zwischen Kellerautomaten und kontextfreien Grammatiken <ul style="list-style-type: none"><li>kontextfreie Grammatik: Definition Anwendungen</li><li>Modell des Kellerautomaten Definition Darstellungsformen Anwendungen / Sprache eines Kellerautomaten</li><li>Zusammenhang zwischen Kellerautomaten/kontextfreien Grammatiken</li><li>Grenzen der Kellerautomaten</li></ul>	Die Schülerinnen und Schüler <ul style="list-style-type: none"><li>analysieren und erläutern die Eigenschaften endlicher Automaten und Kellerautomaten einschließlich ihres Verhaltens bei bestimmten Eingaben,</li><li>ermitteln die Sprache, die ein endlicher Automat oder ein Kellerautomat akzeptiert,</li><li>entwickeln und modifizieren zu einer Problemstellung endliche Automaten oder Kellerautomaten,</li><li>entwickeln zur Grammatik einer regulären oder kontextfreien Sprache einen zugehörigen endlichen Automaten oder einen Kellerautomaten,</li><li>analysieren und erläutern Grammatiken regulärer und kontextfreier Sprachen,</li><li>modifizieren Grammatiken regulärer und kontextfreier Sprachen,</li><li>entwickeln zu einer regulären oder kontextfreien Sprache eine Grammatik, die die Sprache erzeugt,</li><li>beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken.</li></ul>	Arbeitsblätter Orientierungsrahmen: Chomsky Implementierung eines Klammerprüfers

## Unterrichtsvorhaben Q2-II

Thema: Schritte eines Compilers einer formalen Sprache

Leitfragen: Wie funktioniert ein Compiler genau? Wie erkennt ein Compiler die Bestandteile einer Programmiersprache? Wie erkennt er die syntaktische Korrektheit?

Inhaltsfelder: Inhaltsfeld 1 (Daten und ihre Strukturierung), Inhaltsfeld 2 (Algorithmen), Inhaltsfeld 3 (Formale Sprachen und Automaten), Inhaltsfeld 4 (Informatiksysteme)

Absprachen zur vorhabenbezogenen Konkretisierung:

Ausgehend von einer einfachen formalen Sprache (z. B. reduziertes Java) werden die Bestandteile eines Compilers dargestellt:

Der Scanner eines Compilers wird in Form eines endlichen Automaten modelliert und implementiert. Die Begriffe der Symboltabelle und der Tokenliste werden inhaltlich gefüllt. Der Sprachumfang der einfachen formalen Sprache wird leicht erweitert und die Auswirkungen auf den Automaten und die Implementierung wird beobachtet.

Der Parser eines Compilers wird in Form einer kontextfreien Grammatik modelliert und implementiert. Der Sprachumfang der einfachen formalen Sprache wird um weitere Regeln ergänzt und der Parser wird angepasst.

Bei Bedarf wird ein Übersetzer-Modul entwickelt, welches die einfache formale Sprache in eine Sprachebene übersetzt (z. B. interpretiert und compiliert).

Lernmittel / Materialien:

- Arbeitsblätter

Zeitbedarf:10 Stunden

### Sequenzierung des Unterrichtsvorhabens:

<p>Die Schritte eines Compilers</p> <ul style="list-style-type: none"><li>• Scanner: endlicher Automat als Grundlage Symboltabelle und Tokenliste zur Verwaltung und Erkennung des Quelltextes Erweiterung des terminalen Alphabets der zu compilierenden formalen Sprache Implementierung als endlicher Automat</li><li>• Parser: kontextfreie Grammatik als Grundlage Erweiterung des Sprachumfangs Implementierung als kontextfreie Grammatik</li></ul>	<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,</li><li>• stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau,</li><li>• modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten,</li><li>• modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren,</li><li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen lineare und nichtlineare Datensammlungen zu,</li><li>• verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie,</li><li>• ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu,</li><li>• stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar,</li><li>• analysieren und erläutern objektorientierte Modellierungen,</li><li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken,</li><li>• analysieren und erläutern Algorithmen und Programme,</li><li>• modifizieren Algorithmen und Programme,</li></ul>	<p>Arbeitsblätter und Skript zum Compilerbau Mögliche Vertiefung:</p> <ul style="list-style-type: none"><li>• Übersetzer (Überführung in eine Sprachebene)</li></ul>
--	--	--



	<ul style="list-style-type: none"><li>• entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“,</li><li>• implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen,</li><li>• testen Programme systematisch anhand von Beispielen.</li><li>• analysieren und erläutern die Eigenschaften endlicher Automaten und Kellerautomaten einschließlich ihres Verhaltens bei bestimmten Eingaben,</li><li>• ermitteln die Sprache, die ein endlicher Automat oder ein Kellerautomat akzeptiert,</li><li>• entwickeln und modifizieren zu einer Problemstellung endliche Automaten oder Kellerautomaten,</li><li>• stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform,</li><li>• analysieren und erläutern Grammatiken regulärer und kontextfreier Sprachen,</li><li>• modifizieren Grammatiken regulärer und kontextfreier Sprachen,</li><li>• ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird,</li><li>• entwickeln zu einer regulären oder kontextfreien Sprache eine Grammatik, die die Sprache erzeugt,</li><li>• modellieren und implementieren Scanner, Parser und Interpreter zu einer gegebenen regulären Sprache.</li></ul>	
--	--	--

## 2.2. Grundsätze der fachmethodischen und fachdidaktischen Arbeit

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

### Überfachliche Grundsätze:

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

### Fachliche Grundsätze:

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

## 2.3. Grundsätze der Leistungsbewertung und Leistungsrückmeldung

**Hinweis:** Sowohl die Schaffung von **Transparenz bei Bewertungen** als auch die Vergleichbarkeit

von Leistungen sind das Ziel, innerhalb der gegebenen Freiräume Vereinbarungen zu Bewertungskriterien und deren Gewichtung zu treffen.

Auf der Grundlage von §13 - §16 der APO-GOST sowie Kapitel 3 des Kernlehrplans Informatik für die gymnasiale Oberstufe hat die Fachkonferenz des Heinrich-Mann-Gymnasiums im Einklang mit dem entsprechenden schulbezogenen Konzept die nachfolgenden Grundsätze zur Leistungsbewertung und Leistungsrückmeldung beschlossen. Die nachfolgenden Absprachen stellen die Minimalanforderungen an das lerngruppenübergreifende gemeinsame Handeln der Fachgruppenmitglieder dar. Bezogen auf die einzelne Lerngruppe kommen ergänzend weitere der in den Folgeabschnitten genannten Instrumente der Leistungsüberprüfung zum Einsatz.

### 2.3.1. Beurteilungsbereich Klausuren

#### **Verbindliche Absprachen:**

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

#### **Instrumente:**

- Einführungsphase: 1 Klausur je Halbjahr  
Dauer der Klausur: 90 Minuten
- Grund- und Leistungskurse Q 1: 2 Klausuren je Halbjahr  
Dauer der Klausuren: 90 Minuten Unterrichtsstunden (Grundkurs), 135 Minuten (Leistungskurs)
- Grund- und Leistungskurse Q 2.1: 2 Klausuren  
Dauer der Klausuren: 135 Minuten Unterrichtsstunden (Grundkurs), 180 Minuten (Leistungskurs)
- Grund- und Leistungskurse Q 2.2: 1 Klausur unter Abiturbedingungen
- Anstelle einer Klausur kann gemäß dem Beschluss der Lehrerkonferenz in Q 1.2 eine Facharbeit geschrieben werden.

Die Aufgabentypen, sowie die Anforderungsbereiche I-III sind entsprechend den Vorgaben in Kapitel 3 des Kernlehrplans zu beachten.

## Kriterien

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

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

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

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

### 2.3.2. Beurteilungsbereich Sonstige Mitarbeit

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

#### Leistungsaspekte

##### Mündliche Leistungen

- Beteiligung am Unterrichtsgespräch
- Zusammenfassungen zur Vor- und Nachbereitung des Unterrichts
- Präsentation von Arbeitsergebnissen
- Referate
- Mitarbeit in Partner-/Gruppenarbeitsphasen

##### Praktische Leistungen am Computer

- Implementierung, Test und Anwendung von Informatiksystemen

##### Sonstige schriftliche Leistungen

- Arbeitsmappe und Arbeitstagebuch zu einem durchgeführten Unterrichtsvorhaben
- Lernerfolgsüberprüfung durch kurze schriftliche Übungen  
In Kursen, in denen höchstens 50% der Kursmitglieder eine Klausur schreiben, finden schriftliche Übungen mindestens einmal pro Kurshalbjahr statt, in anderen Kursen entscheidet über die Durchführung die Lehrkraft.  
Schriftliche Übung dauern ca. 20 Minuten und umfassen den Stoff der letzten ca. 4–6 Stunden.
- Bearbeitung von schriftlichen Aufgaben im Unterricht

## Kriterien

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

Die Bewertungskriterien stützen sich auf

- die Qualität der Beiträge,
- die Quantität der Beiträge und
- die Kontinuität der Beiträge.

Besonderes Augenmerk ist dabei auf

- die sachliche Richtigkeit,
- die angemessene Verwendung der Fachsprache,
- die Darstellungskompetenz,
- die Komplexität und den Grad der Abstraktion,
- die Selbstständigkeit im Arbeitsprozess,
- die Präzision und
- die Differenziertheit der Reflexion zu legen.

Bei Gruppenarbeiten auch auf

- das Einbringen in die Arbeit der Gruppe,
- die Durchführung fachlicher Arbeitsanteile und
- die Qualität des entwickelten Produktes.

Bei Projektarbeit darüber hinaus auf

- die Dokumentation des Arbeitsprozesses,
- den Grad der Selbstständigkeit,
- die Reflexion des eigenen Handelns und
- die Aufnahme von Beratung durch die Lehrkraft.

## Grundsätze der Leistungsrückmeldung und Beratung

Die Grundsätze der Leistungsbewertung werden zu Beginn eines jeden Halbjahres den Schülerinnen und Schülern transparent gemacht.

Leistungsrückmeldungen können erfolgen

- nach einer mündlichen Überprüfung,
- bei Rückgabe von schriftlichen Leistungsüberprüfungen,
- nach Abschluss eines Projektes,
- nach einem Vortrag oder einer Präsentation,
- bei auffälligen Leistungsveränderungen,
- auf Anfrage,
- als Quartalsfeedback und
- zu Eltern- oder Schülersprechtagen.

Die Leistungsrückmeldung kann

- durch ein Gespräch mit der Schülerin oder dem Schüler,
- durch einen Feedbackbogen,
- durch die schriftliche Begründung einer Note oder
- durch eine individuelle Lern-/Förderempfehlung

erfolgen.

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