

# **Schulinterner Lehrplan des Ravensberger Gymnasiums zum Kernlehrplan für die gymnasiale Oberstufe**

## **Informatik**

**(Stand: 12.12.2018)**

# Inhalt

<b>1 Die Fachgruppe Informatik am Ravensberger Gymnasium Herford</b>	<b>3</b>
<b>2 Entscheidungen zum Unterricht</b>	<b>5</b>
2.1 Unterrichtsvorhaben	5
2.1.1 <i>Übersichtsraster Unterrichtsvorhaben</i>	6
2.1.2 <i>Konkretisierte Unterrichtsvorhaben</i>	17
2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit	64
2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung	66
2.3.1 <i>Beurteilungsbereich Klausuren</i>	66
2.3.2 <i>Beurteilungsbereich Sonstige Mitarbeit</i>	67
<b>3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen</b>	<b>70</b>
<b>4 Qualitätssicherung und Evaluation</b>	<b>71</b>

# **1 Die Fachgruppe Informatik am Ravensberger Gymnasium Herford**

Beim Ravensberger Gymnasium handelt es sich um eine im Wesentlichen dreizügige Schule im Zentrum von Herford mit zurzeit ca. 730 Schülerinnen und Schülern und ca. 70 Lehrerinnen und Lehrern. Das Einzugsgebiet der Schule umfasst die Herforder Innenstadt sowie umliegende Städte und Gemeinden, was zum Teil auf das Angebot der Schule im Fach Informatik zurückzuführen ist. Im Bereich der Sekundarstufe II kooperiert das Ravensberger Gymnasium mit dem Friedrichs-Gymnasium als direkter Nachbarschule und bietet mit ihm zahlreiche gemeinsame Kurse an, unter anderem in Informatik.

Schülerinnen und Schüler aller Klassen- und Jahrgangsstufen werden zur Teilnahme an Wettbewerben im Fach Informatik angehalten. Regelmäßig belegt das RGH deutschlandweit vordere Plätze beim Informatik-Biber und beim Bundeswettbewerb Informatik.

In der Jahrgangsstufen 5 werden informatische Grundlagen insbesondere im Umgang mit informatischen Systemen im Rahmen von IKG I vermittelt. In Jahrgangsstufe 7 liegt im Rahmen von IKG II der Schwerpunkt auf einem verantwortungsbewussten Umgang mit persönlichen Daten in vernetzten Systemen.

Das Fach Informatik wird am Ravensberger Gymnasium ab der Jahrgangsstufe 8 im Differenzierungsbereich dreistündig unterrichtet und von etwa einem Viertel der Schülerinnen und Schüler besucht. In der zweijährigen Laufzeit dieser Kurse wird in altersstufengerechter Weise unter anderem auf Grundlagen der Algorithmik am Beispiel einer didaktischen Lernumgebung, auf die technische Informatik am Beispiel von Mikrocontrollern und auf Robotik eingegangen.

In der Sekundarstufe II bietet das Ravensberger Gymnasium für die eigenen Schülerinnen und Schüler in allen Jahrgangsstufen jeweils einen Grundkurs und in der Qualifikationsphase jeweils einen Leistungskurs in Informatik an.

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

Der Unterricht der Sekundarstufe II wird mit Hilfe der Programmiersprache Java durchgeführt.

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

Die gemeinsame Entwicklung von Materialien und Unterrichtsvorhaben, die Evaluation von Lehr- und Lernprozessen sowie die stetige Überprüfung und eventuelle Modi-

fikation des schulinternen Curriculums durch die Fachkonferenz Informatik stellen einen wichtigen Beitrag zur Qualitätssicherung und -entwicklung des Unterrichts dar.

Zurzeit besteht die Fachschaft Informatik des Ravensberger Gymnasiums aus drei Lehrkräften, denen zwei Computerräume mit je 15 Computerarbeitsplätzen und ein Selbstlernzentrum mit 15 Plätzen zur Verfügung stehen. Alle Arbeitsplätze sind an das schulinterne Rechnernetz angeschlossen, so dass Schülerinnen und Schüler über einen individuell gestaltbaren Zugang zum zentralen Server der Schule alle Arbeitsplätze der drei Räume zum Zugriff auf ihre eigenen Daten, zur Recherche im Internet oder zur Bearbeitung schulischer Aufgaben verwenden können.

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

## **2 Entscheidungen zum Unterricht**

### **2.1 Unterrichtsvorhaben**

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

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

<b>Einführungsphase (84 Stunden)</b>	
<p><u>Unterrichtsvorhaben I:</u></p> <p><b>Thema:</b> <i>Einordnung der Informatik in den gesellschaftlichen Kontext</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>• Argumentieren</li> <li>• Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>• Informatik, Mensch und Gesellschaft</li> <li>• Informatiksysteme</li> </ul> <p><b>Inhaltlicher Schwerpunkt:</b></p> <ul style="list-style-type: none"> <li>• Geschichtlicher Abriss</li> <li>• Wirkung der Automatisierung</li> <li>• Aufbau von Einzelrechnern</li> <li>• Digitalisierung</li> <li>• Dateisystem und Internet</li> </ul> <p><b>Zeitbedarf:</b> 15 Std.</p>	<p><u>Unterrichtsvorhaben II:</u></p> <p><b>Thema:</b> <i>Grundlagen der objektorientierten Programmierung und algorithmische Grundstrukturen</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>• Implementieren</li> <li>• Modellieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>• Algorithmen</li> <li>• Formale Sprachen und Automaten</li> <li>• Daten und ihre Strukturierung</li> </ul> <p><b>Inhaltlicher Schwerpunkt:</b></p> <ul style="list-style-type: none"> <li>• Analyse, Entwurf und Implementierung einfacher Algorithmen</li> <li>• Syntax und Semantik einer Programmiersprache</li> <li>• Objekte und Klassen</li> </ul> <p><b>Zeitbedarf:</b> 30 Std.</p>
<p><u>Unterrichtsvorhaben III:</u></p> <p><b>Thema:</b> <i>Entwurf und Implementierung alltagstauglicher Anwendungsprogramme</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>• Modellieren</li> <li>• Implementieren</li> <li>• Darstellen und Interpretieren</li> <li>• Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>• Daten und ihre Strukturierung</li> <li>• Algorithmen</li> <li>• Informatik, Mensch und Gesellschaft</li> </ul> <p><b>Inhaltlicher Schwerpunkt:</b></p> <ul style="list-style-type: none"> <li>• Objekte und Klassen</li> <li>• Komplexere Algorithmen</li> <li>• Einsatz von Informatiksystemen</li> </ul> <p><b>Zeitbedarf:</b> 24 Std.</p>	<p><u>Unterrichtsvorhaben IV:</u></p> <p><b>Thema:</b> <i>Ordnung herstellen – Suchen und Sortieren</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>• Implementieren</li> <li>• Argumentieren</li> <li>• Modellieren</li> <li>• Darstellen und Interpretieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>• Algorithmen</li> <li>• Objekte und Klassen</li> </ul> <p><b>Inhaltlicher Schwerpunkt:</b></p> <ul style="list-style-type: none"> <li>• Analyse, Entwurf und Implementierung von Such- und Sortieralgorithmen</li> </ul> <p><b>Zeitbedarf:</b> 15 Std.</p>

## Übersicht über die Unterrichtsvorhaben

E-Phase		
Unterrichtsvorhaben	Thema	Stundenzahl
I	s.o.	15
II	s.o.	30
III	s.o.	24
IV	s.o.	15
	Summe:	84

## II) Qualifikationsphase (Q1 und Q2) - GRUNDKURS

Qualifikationsphase 1	
<p><u>Unterrichtsvorhaben Q1-I</u></p> <p><b>Thema:</b> <i>Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>• Argumentieren</li> <li>• Modellieren</li> <li>• Implementieren</li> <li>• Darstellen und Interpretieren</li> <li>• Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>• Daten und ihre Strukturierung</li> <li>• Algorithmen</li> <li>• Formale Sprachen und Automaten</li> <li>• Informatiksysteme</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>• Objekte und Klassen</li> <li>• Analyse, Entwurf und Implementierung von Algorithmen</li> <li>• Syntax und Semantik einer Programmiersprache</li> <li>• Nutzung von Informatiksystemen</li> </ul> <p><b>Zeitbedarf:</b> 9 Stunden</p>	<p><u>Unterrichtsvorhaben Q1-II</u></p> <p><b>Thema:</b> <i>Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>• Argumentieren</li> <li>• Modellieren</li> <li>• Implementieren</li> <li>• Darstellen und Interpretieren</li> <li>• Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>• Daten und ihre Strukturierung</li> <li>• Algorithmen</li> <li>• Formale Sprachen und Automaten</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>• Objekte und Klassen</li> <li>• Analyse, Entwurf und Implementierung von Algorithmen</li> <li>• Algorithmen in ausgewählten informatischen Kontexten</li> <li>• Syntax und Semantik einer Programmiersprache</li> </ul> <p><b>Zeitbedarf:</b> 18 Stunden</p>



Unterrichtsvorhaben Q1-III

**Thema:**

Suchen und Sortieren auf linearen Datenstrukturen

**Zentrale Kompetenzen:**

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

**Inhaltsfelder:**

- Algorithmen
- Formale Sprachen und Automaten

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 15 Stunden

Unterrichtsvorhaben Q1-IV

**Thema:**

Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen

**Zentrale Kompetenzen:**

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

**Inhaltsfelder:**

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

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 24 Stunden

## Qualifikationsphase 1

### Unterrichtsvorhaben Q1-V

**Thema:**

Sicherheit und Datenschutz in Netzstrukturen

**Zentrale Kompetenzen:**

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

**Inhaltsfelder:**

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 15 Stunden

**Summe Qualifikationsphase 1: 81 Stunden**

## Qualifikationsphase 2

### Unterrichtsvorhaben Q2-I

**Thema:**

Endliche Automaten und formale Sprachen und Grenzen der Automatisierbarkeit

**Zentrale Kompetenzen:**

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

**Inhaltsfelder:**

- Endliche Automaten und formale Sprachen
- Informatik, Mensch und Gesellschaft

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 20 Stunden

### Unterrichtsvorhaben Q2-II

**Thema:**

Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

**Zentrale Kompetenzen:**

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

**Inhaltsfelder:**

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

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 26 Stunden

<b>Qualifikationsphase 2</b>	
<p><u>Unterrichtsvorhaben Q2-III</u></p> <p><b>Thema:</b> Prinzipielle Arbeitsweise eines Computers</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>• Argumentieren</li> <li>• Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>• Informatiksysteme</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>• Einzelrechner und Rechnernetzwerke</li> </ul> <p><b>Zeitbedarf:</b> 6 Stunden</p>	
<b>Summe Qualifikationsphase 2: 52 Stunden</b>	

### III) Qualifikationsphase (Q1 und Q2) – LEISTUNGSKURS

<b>Qualifikationsphase 1</b>	
<p><u>Unterrichtsvorhaben Q1-I</u></p> <p><b>Thema:</b> <i>Wiederholung der objektorientierten Modellierung und Programmierung</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>• Argumentieren</li> <li>• Modellieren</li> <li>• Implementieren</li> <li>• Darstellen und Interpretieren</li> <li>• Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>• Daten und ihre Strukturierung</li> <li>• Algorithmen</li> <li>• Formale Sprachen und Automaten</li> <li>• Informatiksysteme</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>• Objekte und Klassen</li> <li>• Analyse, Entwurf und Implementierung von Algorithmen</li> <li>• Syntax und Semantik einer Programmiersprache</li> <li>• Nutzung von Informatiksystemen</li> </ul> <p><b>Zeitbedarf:</b> 15 Stunden</p>	<p><u>Unterrichtsvorhaben Q1-II</u></p> <p><b>Thema:</b> Modellierung und Implementierung von dynamischen linearen Datenstrukturen und von Anwendungen mit dynamischen linearen Datenstrukturen in kontextbezogenen Problemstellungen</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>• Argumentieren</li> <li>• Modellieren</li> <li>• Implementieren</li> <li>• Darstellen und Interpretieren</li> <li>• Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>• Daten und ihre Strukturierung</li> <li>• Algorithmen</li> <li>• Formale Sprachen und Automaten</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>• Objekte und Klassen</li> <li>• Analyse, Entwurf und Implementierung von Algorithmen</li> <li>• Algorithmen in ausgewählten informatischen Kontexten</li> <li>• Syntax und Semantik einer Programmiersprache</li> </ul> <p><b>Zeitbedarf:</b> 25 Stunden</p>

Unterrichtsvorhaben Q1-III

**Thema:**

Suchen und Sortieren auf linearen Datenstrukturen sowie Beurteilung der Verfahren hinsichtlich ihrer Komplexitätsklassen.

**Zentrale Kompetenzen:**

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

**Inhaltsfelder:**

- Algorithmen
- Formale Sprachen und Automaten

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 20 Stunden

Unterrichtsvorhaben Q1-IV

**Thema:**

Modellierung und Implementierung von dynamischen **nicht**-linearen Datenstrukturen und von Anwendungen mit dynamischen nicht-linearen Datenstrukturen in kontextbezogenen Problemstellungen.

**Zentrale Kompetenzen:**

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

**Inhaltsfelder:**

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

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 40 Stunden

## Qualifikationsphase 1

### Unterrichtsvorhaben Q1-V

**Thema:**

Grundlagen der Netzwerkkommunikation sowie Modellierung und Implementierung von Client-Server-Anwendungen in kontextbezogenen Problemstellungen sowie Sicherheit in Informatiksystemen

**Zentrale Kompetenzen:**

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

**Inhaltsfelder:**

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 24 Stunden

**Summe Qualifikationsphase 1: 124 Stunden**

## Qualifikationsphase 2

Unterrichtsvorhaben Q2-I

**Thema:**

Endliche Automaten und formale Sprachen  
und Grenzen der Automatisierbarkeit

**Zentrale Kompetenzen:**

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

**Inhaltsfelder:**

- Automaten und formale Sprachen
- Informatik, Mensch und Gesellschaft

**Inhaltliche Schwerpunkte:**

- Endliche Automaten und Kellerauto-  
maten
- Grammatiken regulärer und kontext-  
freier Sprachen
- Scanner, Parser und Interpreter
- Möglichkeiten und Grenzen von Au-  
tomaten und formalen Sprachen
- Grenzen der Automatisierung

**Zeitbedarf:** 27 Stunden

Unterrichtsvorhaben Q2-II

**Thema:**

Prinzipielle Arbeitsweise eines Computers

**Zentrale Kompetenzen:**

- Argumentieren
- Kommunizieren und Kooperieren

**Inhaltsfelder:**

- Informatiksysteme

**Inhaltliche Schwerpunkte:**

- Einzelrechner und Rechnernetzwerke

**Zeitbedarf:** 10 Stunden

## Qualifikationsphase 2

### Unterrichtsvorhaben Q2-III

**Thema:**

Modellierung, Implementierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

**Zentrale Kompetenzen:**

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

**Inhaltsfelder:**

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

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 40 Stunden

### Unterrichtsvorhaben Q2-IV

**Thema:**

Umsetzung eines selbstgewählten Projekts – Durchführung eines vollständigen Softwareentwicklungszyklus

**Zentrale Kompetenzen:**

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

**Inhaltsfelder:**

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

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 20 Stunden

**Summe Qualifikationsphase 2: 97 Stunden**



## 2.1.2 Konkretisierte Unterrichtsvorhaben

### I) Einführungsphase

#### **UV1 Thema:** *Einordnung der Informatik in den gesellschaftlichen Kontext*

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
1. Geschichtlicher Abriss <ul style="list-style-type: none"><li>- Überblick über die Entwicklung der Fachwissenschaft Informatik und deren Teilgebiete</li><li>- Identifikation von Einsatzbereichen der EDV - Auswirkungen auf die Gesellschaft</li></ul>	Die Schülerinnen und Schüler <ul style="list-style-type: none"><li>- erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A)</li><li>- bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A)</li><li>- beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der "von-Neumann-Architektur" (A)</li><li>- nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K)</li><li>- stellen ganze Zahlen und Zeichen in Binärcodes dar (D)</li><li>- interpretieren Binärcodes als ganze Zahlen und Zeichen</li><li>- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung</li></ul>
2. Aufbau informatischer Systeme <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>	
3. Information und deren Codierung und Speicherung <ul style="list-style-type: none"><li>- Informatik als Wissenschaft der Verarbeitung von Information</li><li>- Darstellung von Informationen durch Binärzahlen</li><li>- Speichern von Daten mit informatischen Systemen am Beispiel der Schulrechner</li></ul>	

---

<p>4. Datenschutz an verschiedenen Fallbeispielen</p>	
---	--

- Erarbeitung grundlegender Begriffe des Datenschutzes
- Diskussion und Bewertung von Fallbeispielen aus dem Themenbereich Datenschutz (z.B. anhand des BigBrother Awards)

**UV2 Thema: Grundlagen der objektorientierten Programmierung und algorithmische Grundstrukturen anhand des Java-Hamsters**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Identifikation von Objekten</p> <ul style="list-style-type: none"> <li>- Objekte an Beispielen aus der Umwelt</li> <li>- Zusammenfassung von Objekten zu Klassen</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</li> <li>- modellieren Klassen mit ihren Attributen und ihren Methoden</li> <li>- entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M)</li> <li>- implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I)</li> <li>- implementieren Algorithmen unter Beachtung der Syntax und Semantik der Programmiersprache Java (I)</li> <li>- interpretieren Fehlermeldungen und korrigieren den Quellcode (I)</li> <li>- testen Programme schrittweise anhand von Beispielen (I)</li> <li>- modellieren Klassen unter Verwendung von Vererbung (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> </ul>
<p>2. Virtuelle Lebewesen</p> <ul style="list-style-type: none"> <li>- Steuerung des Java-Hamsters mittels der Programmiersprache Java</li> <li>- Identifikation der Freunde des Hamsters als Objekte einer Klasse mit unterschiedlichen Ausprägungen</li> <li>- Erweiterung des Methodenumfangs (Prozeduren) mit Hilfe von Unterklassen und Verdeutlichung des Prinzips der Vererbung</li> <li>- Einführung von Variablen, einfachen Datentypen und Kontrollstrukturen sowie Funktionen</li> <li>- Implementierung eigener Methoden mit und ohne Parameterübergabe</li> <li>- Implementierung der Kommunikation zwischen Objekten</li> <li>- Darstellung von Programmabläufen mit Hilfe von Struktogrammen</li> </ul>	

### 3. Objektorientierte Modellierung am Beispiel des Hamsters

- Die Standardklasse Hamster im Klassendiagramm
- Darstellung der Erweiterung um Unterklassen der Standardklasse
- Identifikation unterschiedlicher Zugriffsrechte für Klassen, Attribute und Methoden und Darstellung im Diagramm
- Thematisierung des Geheimnisprinzips

**UV3 Thema: Entwurf und Implementierung alltagstauglicher Anwendungsprogramme unter Verwendung des java-Editors**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Analyse verschiedener komplexer Problemstellungen im Sachzusammenhang</p> <ul style="list-style-type: none"> <li>- Modellierung der Problemstellungen im Entwurfs- und Implementationsdiagramm</li> <li>- Modellierung einer geeigneten Benutzeroberfläche</li> <li>- Identifikation verschiedener Anwendungsfälle und ggf. Darstellung im Use-Case-Diagramm</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 den Zustand eines Objekts dar (D)</li> <li>- stellen die Kommunikation zwischen Objekten grafisch dar (M)</li> <li>- stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D)</li> <li>- analysieren und erläutern eine objektorientierte Modellierung (A)</li> <li>- dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D)</li> <li>- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I) nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D)</li> </ul>
<p>2. Analyse, Entwurf und Implementierung verschiedener Prüfprogramme unter Verwendung linearer Datenstrukturen wie Array und Stack sowie der Methoden der Klassen String und Character</p>	

---

## **UV4 Thema: Ordnung herstellen - Suchen und Sortieren**

### **Unterrichtssequenzen**

1. Die Lineare Suche als einfacher Suchalgorithmus
  - Entwurf eines Algorithmus zur linearen Suche
  - Implementierung auf einer linearen Datensammlung (Array)
2. Sortierverfahren
  - Analyse, Erläuterung und Anwendung des Bubblesorts als Beispiel eines einfachen Sortierverfahrens
  - Entwurf und ggf. Implementierung eines weiteren einfachen Sortierverfahrens durch Modifizierung des Bubblesorts
  - Analyse des Rechenaufwandes bei einfachen Sortierverfahren in Abhängigkeit der Datenmenge

### **Zu entwickelnde Kompetenzen**

- Die Schülerinnen und Schüler
- analysieren und erläutern einfache Algorithmen und Programme (A)
  - entwerfen Suchalgorithmen und wenden sie auf Beispiele an (D)
  - analysieren Sortieralgorithmen und wenden sie auf Beispiele an (D)
  - entwerfen einen weiteren Algorithmus zum Sortieren (M)
  - modifizieren einfache Algorithmen und Programme (I)
  - beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeitaufwand und Speicherplatzbedarf (A)

---

## II) Qualifikationsphase - GRUNDKURS

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

Die Schülerinnen und Schüler

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

## Unterrichtsvorhaben Q1-I:

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

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

**Zeitbedarf:** 9 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><b>Wiederholung und Erweiterung der objektorientierten Modellierung und Programmierung durch Analyse und Erweiterung eines kontextbezogenen Beispiels</b></p> <ul style="list-style-type: none"><li>(a) Analyse der Problemstellung</li><li>(b) Analyse der Modellierung (Implementationsdiagramm)</li><li>(c) Erweiterung der Modellierung im Implementationsdiagramm (Vererbung, abstrakte Klasse)</li><li>(d) Kommunikation zwischen mindestens zwei Objekten (grafische Darstellung)</li><li>(e) Dokumentation von Klassen</li><li>(f) Implementierung der Anwendung oder von Teilen der Anwendung</li></ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"><li>• <b>analysieren und erläutern objektorientierte Modellierungen (A),</b></li><li>• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li><li>• <b>modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M),</b></li><li>• <b>ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M),</b></li><li>• <b>modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),</b></li></ul>	<p><i>Beispiel für eine Vorgehensweise:</i> Medien-Datenbank</p> <ul style="list-style-type: none"><li>- Vertiefung Grundprinzipien OOM</li><li>- Dokumentation von Quellcode</li><li>- Objekt- &amp; Implementationsdiagramme</li><li>- Casten, Subtyping, Polymorphie</li><li>- Abstrakte Klassen und Methoden am Bsp. der Klasse Medium</li></ul> <p><i>Medien:</i> Java-Editor</p>



	<ul style="list-style-type: none"><li>• <b>implementieren Klassen in einer Programmiersprache (I),</b></li><li>• nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li><li>• wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I),</li><li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li><li>• <b>stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D),</b></li><li>• <b>dokumentieren Klassen (D),</b></li><li>• <b>stellen die Kommunikation zwischen Objekten grafisch dar (D).</b></li></ul>	
--	---	--

## Unterrichtsvorhaben Q1-II:

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

**Leitfrage:** *Wie kann eine uns vorher unbekannte Anzahl an Daten verwaltet werden?*

**Zeitbedarf:** 18 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><b>1. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse <code>Queue</code></b></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse <code>Queue</code></p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse <code>Queue</code></p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"><li>• <b>erläutern Operationen dynamischer (linearer) Datenstrukturen (A),</b></li><li>• analysieren und erläutern Algorithmen und Programme (A),</li><li>• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li><li>• <b>ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare Datensammlungen zu (M),</b></li><li>• ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li><li>• modifizieren Algorithmen und Programme (I),</li><li>• <b>implementieren iterative auch unter Verwendung von dynamischen Daten-</b></li></ul>	<p><i>Beispiel:</i> Patientenwarteschlange</p>
<p><b>2. Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse <code>Stack</code></b></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse <code>Stack</code></p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines</p>	<ul style="list-style-type: none"><li>• <b>implementieren iterative auch unter Verwendung von dynamischen Daten-</b></li></ul>	<p><i>Beispiel:</i> Palindrom</p>

<p>oder mehrerer Objekte der Klasse Stack</p>	<p><b>strukturen (I),</b></p> <ul style="list-style-type: none"> <li>• nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> </ul>	
<p><b>3. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse List</b></p> <p>(a) Erarbeitung der Vorteile der Klasse List im Gegensatz zu den bereits bekannten linearen Strukturen</p> <p>(b) Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse List.</p>	<ul style="list-style-type: none"> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>• testen Programme systematisch anhand von Beispielen (I),</li> <li>• <b>stellen lineare Strukturen grafisch dar und erläutern ihren Aufbau (D).</b></li> </ul>	<p><i>Beispiel:</i> Einkaufsliste</p>

## Unterrichtsvorhaben Q1-III:

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

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

**Zeitbedarf:** 15 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><b>1. Sortieren in Listen und Arrays - Entwicklung bzw. Implementierung von Sortierverfahren</b></p> <p>(a) Entwicklung und Implementierung eines einfachen Sortierverfahrens für ein Feld</p> <p>(b) Anwendung verschiedener Sortierverfahren und Veranschaulichung an Zahlenbeispielen</p> <p>(c) Implementierung eines einfachen Sortierverfahrens für eine Liste</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• analysieren und erläutern Algorithmen und Programme (A),</li> <li>• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li> <li>• <b>beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A),</b></li> <li>• <b>entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M),</b></li> </ul>	<p><i>Beispiel:</i> Karteiverwaltung</p> <ul style="list-style-type: none"> <li>- Bubblesort</li> <li>- Sortieren durch direktes Einfügen (Insertion Sort)</li> <li>- Sortieren durch direktes Auswählen (Selection Sort)</li> <li>- Ggf. Quicksort</li> </ul>
<p><b>2. Untersuchung der Effizienz der verschiedenen Sortierverfahren auf linearen Listen</b></p> <p>(a) Untersuchung der Anzahl der Vergleichsoperationen und des Speicherbedarf bei den Sortierverfahren</p> <p>(b) Beurteilung der Effizienz der Sortierverfahren</p>	<ul style="list-style-type: none"> <li>• modifizieren Algorithmen und Programme (I),</li> <li>• <b>implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),</b></li> <li>• <b>implementieren und erläutern iterative</b></li> </ul>	<ul style="list-style-type: none"> <li>- Effizienz von Algorithmen (Komplexitätsklassen)</li> </ul>

<p><b>3. Suchen von Daten in Listen und Arrays</b></p> <p>(a) Lineare Suche in Listen und in Arrays  (b) Binäre Suche in Arrays als Beispiel für rekursives Problemlösen  (c) Untersuchung der beiden Suchverfahren hinsichtlich ihrer Effizienz (Laufzeitverhalten, Speicherbedarf)</p>	<p><b>und rekursive Such- und Sortierverfahren (I),</b></p> <ul style="list-style-type: none"> <li>• nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>• testen Programme systematisch anhand von Beispielen (I),</li> <li>• <b>stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).</b></li> </ul>	<p>- Iterative und rekursive Implementierung der binären Suche</p> <hr/> <p>- <i>Fibonacci Zahlen, Fakultät, Bild im Bild, Deutsche Sprache</i></p>
<p><b>4. Rekursion</b></p> <p>(a) Das Prinzip der Rekursion an Beispielen aus anderen Fachbereichen (Mathematik, Deutsch, Kunst)  (b) Anwenden und Darstellen eines rekursiven Funktionsaufrufs</p>		

## Unterrichtsvorhaben Q1-IV:

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

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

**Zeitbedarf:** 24 Stunden

### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><b>1. Analyse von Baumstrukturen in verschiedenen Kontexten</b></p> <p>(a) Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit)</p> <p>(b) Aufbau und Darstellung von binären Bäumen</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• <b>erläutern Operationen dynamischer (nichtlinearer) Datenstrukturen (A),</b></li> <li>• analysieren und erläutern Algorithmen und Programme (A),</li> <li>• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li> </ul>	<p><i>Beispiel: Stammbaum</i></p>
<p><b>2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse <code>BinaryTree</code></b></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms</p> <p>(c) Erarbeitung der Klasse <code>BinaryTree</code> und beispielhafte Anwendung der</p>	<ul style="list-style-type: none"> <li>• ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li> <li>• <b>ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie nichtlineare Datensammlungen zu (M),</b></li> <li>• modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren</li> </ul>	<p><i>Beispiel: Morsebaum</i></p> <ul style="list-style-type: none"> <li>- Binärbaum am Bsp. Morsebaum (Codierung): rekursiver Aufbau</li> <li>- Decodierung</li> <li>- Traversierungsalgorithmen pre-, post-, inorder</li> <li>- Codierung</li> </ul>

<p>Operationen</p> <p>(d) Implementierung der Anwendung oder von Teilen der Anwendung</p> <p>(e) Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf</p>	<p>und Generalisieren (M),</p> <ul style="list-style-type: none"> <li>• <b>entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M),</b></li> <li>• <b>implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),</b></li> <li>• modifizieren Algorithmen und Programme (I),</li> <li>• nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>• testen Programme systematisch anhand von Beispielen (I),</li> <li>• <b>stellen nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D),</b></li> <li>• <b>stellen iterative und rekursive Algorithmen umgangssprachlich dar (D).</b></li> </ul>	
<p><b>3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse <code>BinarySearchTree</code></b></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramm, grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften</p> <p>(c) Erarbeitung der Klasse <code>BinarySearchTree</code> und Einführung des Interface <code>Item</code> zur Realisierung einer geeigneten Ordnungsrelation</p> <p>(d) Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums</p>		<p><i>Beispiel:</i> Informatikerbaum als binärer Suchbaum</p> <p><i>Beispiel:</i> Huffman-Codierung (Zeichenhäufigkeit) zur Herleitung des Interface <code>Item</code></p>

## Unterrichtsvorhaben Q1-V:

**Thema:** Sicherheit und Datenschutz in Netzstrukturen und Informatiksystemen

**Leitfragen:** *Wie werden Daten in Netzwerken übermittelt? Was sollte man in Bezug auf die Sicherheit beachten? Welche moralische Verantwortung tragen Informatikerinnen und Informatiker hinsichtlich der gesellschaftlichen Auswirkungen informatischer Systeme?*

**Zeitbedarf:** 15 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><b>Daten in Netzwerken und Sicherheitsaspekte in Netzen sowie beim Zugriff auf Datenbanken</b></p> <p>(a) Beschreibung eines Datenbankzugriffs im Netz anhand eines Anwendungskontextes und einer Client-Server-Struktur zur Klärung der Funktionsweise eines Datenbankzugriffs</p> <p>(b) Netztopologien als Grundlage von Client-Server-Strukturen und TCP/IP-Schichtenmodell als Beispiel für eine Paketübermittlung in einem Netz</p> <p>(c) Vertraulichkeit, Integrität, Authentizität, Verfügbarkeit in Netzwerken sowie symmetrische und asymmetrische kryptografische Verfahren (Cäsar-, Vigenère-, RSA-Verfahren) als Methoden Daten im Netz verschlüsselt zu übertragen</p> <p>(d) Erarbeitung von Interesse</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• <b>beschreiben und erläutern Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A),</b></li> <li>• <b>analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A),</b></li> <li>• <b>untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen (A),</b></li> <li>• <b>untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A),</b></li> <li>• <b>nutzen bereitgestellte Informatiksysteme</b></li> </ul>	<p><i>Beispiel: Kommunikation übers Netz (Goldcity etc.)</i></p> <p><i>Beispiel: Schulnetz und LAN-Party</i></p> <ul style="list-style-type: none"> <li>• <i>Datenübertragung (Paketübermittlung)</i></li> <li>• <i>Netzwerktopologien</i></li> <li>• <i>Protokolle</i></li> <li>• <i>TCP/IP-Schichtenmodell</i></li> </ul> <p><i>Materialien:</i></p> <p><i>Filius, Film „Sendung mit der Maus“</i></p> <p><i>Kryptologie</i></p> <p><i>am Bsp. einer Geburtstagsfeier</i></p> <ul style="list-style-type: none"> <li>• <i>Caesar-Verfahren</i></li> <li>• <i>Vigenère-Verfahren</i></li> <li>• <i>RSA</i></li> <li>• <i>Diffie-Hellmann-Schlüsselaustauschverfahren</i></li> </ul> <p><i>Ethik</i></p> <p><i>am Beispiel der Verantwortungsethik nach</i></p>



<p>verschiedener Interessengruppen im Hinblick auf Fallbeispiele</p> <p>(e) Erarbeitung eines einfachen moralischen Bewertungsmaßstabes</p> <p>(f) Anwendung auf Fallbeispiele</p> <p>(g) Einführung in das Verfassen von Stellungnahmen sowie in Kriterien für gelungene Stellungnahmen</p> <p>(h) Erarbeitung von Stellungnahmen zu Fallbeispielen</p>	<p>und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D).</p>	<p><i>Hans Jonas</i></p> <p><i>Erarbeitung an verschiedenen Fallbeispielen wie bspw. autonomes Fahren, Software für Drohnen, Bewertungssystem für chinesische Bürger</i></p>
--	--	--

## Unterrichtsvorhaben Q2-I:

**Thema:** Endliche Automaten und formale Sprachen

**Leitfragen:** *Wie können endliche Automaten (in alltäglichen Kontexten oder zu informatischen Problemstellungen) modelliert werden? Wie können Sprachen durch Grammatiken beschrieben werden? Welche Zusammenhänge gibt es zwischen formalen Sprachen, endlichen Automaten und regulären Grammatiken? Welche Möglichkeiten bieten Informatiksysteme und wo liegen ihre Grenzen?*

**Zeitbedarf:** 20 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p><b>1. Endliche Automaten</b></p> <p>(a) Von realen Automaten zur formalen Beschreibung eines endlichen Automaten</p> <p>(b) Untersuchung, Darstellung und Entwicklung von Transduktoren, Akzeptoren und nichtdeterministischen endlichen Automaten</p> <p>(c) Umwandlung nichtdeterministischer endlicher Automaten in deterministische endliche Automaten</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• <b>analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens auf bestimmte Eingaben (A),</b></li> <li>• <b>analysieren und erläutern Grammatiken regulärer Sprachen (A),</b></li> <li>• <b>zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A),</b></li> <li>• <b>ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A),</b></li> <li>• <b>entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M),</b></li> <li>• <b>entwickeln und modifizieren zu einer Problemstellung endliche Automaten</b></li> </ul>	<p><i>Beispiele:</i> Ticketautomat der Bahn, Cola-Automat, Geldspielautomat, Akzeptor für bestimmte Zahlenmengen (<math>\{N, Z,  R\}</math>)</p>
<p><b>2. Untersuchung und Entwicklung von Grammatiken regulärer Sprachen</b></p> <p>(a) Erarbeitung der formalen Darstellung regulärer Grammatiken</p> <p>(b) Untersuchung, Modifikation und Entwicklung von Grammatiken</p> <p>(c) Entwicklung von endlichen Automaten</p>	<p>(siehe oben)</p>	<p><i>Beispiele:</i> Die deutsche Grammatik (Satzgliederungsgrammatik), reguläre Grammatik für Wörter, die bestimmte Zahlen repräsentieren,</p>

<p>zum Erkennen regulärer Sprachen die durch Grammatiken gegeben werden (d) Entwicklung regulärer Grammatiken zu endlichen Automaten</p>	<p>(M),</p> <ul style="list-style-type: none"> <li>• entwickeln zur akzeptierten Sprache eines Automaten die zugehörige Grammatik (M),</li> <li>• entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M),</li> <li>• modifizieren Grammatiken regulärer Sprachen (M),</li> </ul>	
<p><b>3. Grenzen endlicher Automaten</b> a) Ausblick: Kellerautomat</p>	<ul style="list-style-type: none"> <li>• entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M),</li> <li>• stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D),</li> <li>• ermitteln die Sprache, die ein endlicher Automat akzeptiert (D).</li> <li>• beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D),</li> <li>• untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A)</li> <li>• nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D).</li> </ul>	<p><i>Beispiele:</i> Klammerausdrücke, <math>a^n b^n</math> im Vergleich zu <math>(ab)^n</math></p>
<p><b>4. Grenzen der Automatisierbarkeit</b> a) Untersuchung verschiedener nicht in polynomialer Zeit berechenbarer Probleme der Informatik b) Vorstellung des Halteproblems c) Unlösbarkeit des Halteproblems d) Beurteilung des Einsatzes von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen</p>	<ul style="list-style-type: none"> <li>• entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M),</li> <li>• stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D),</li> <li>• ermitteln die Sprache, die ein endlicher Automat akzeptiert (D).</li> <li>• beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D),</li> <li>• untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A)</li> <li>• nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D).</li> </ul>	<p><i>Beispiel:</i> Travelling Salesman Problem usw., Halteproblem, Affenpuzzle</p>

## Unterrichtsvorhaben Q2-II:

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

**Leitfragen:** *Wie können Fragestellungen mit Hilfe einer Datenbank beantwortet werden? Wie entwickelt man selbst eine Datenbank für einen Anwendungskontext? Welche datenschutzrechtlichen Vorgaben sind zu beachten?*

**Zeitbedarf:** 26 Stunden

### Sequenzierung des Unterrichtsvorhabens

Anmerkung: Unterrichtssequenz 1 und 2 können in ihrer Reihenfolge auch getauscht werden und greifen in komplexen Beispielen ineinander.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><b>1. Nutzung von relationalen Datenbanken</b></p> <p>(a) Aufbau von Datenbanken und Grundbegriffe</p> <ul style="list-style-type: none"> <li>• Entwicklung von Fragestellungen zur vorhandenen Datenbank</li> <li>• Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, Datenbankschema</li> </ul> <p>(b) SQL-Abfragen</p> <ul style="list-style-type: none"> <li>• Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL (SELECT (DISTINCT) ...FROM, WHERE, AND, OR, NOT) auf einer Tabelle</li> <li>• Analyse und Erarbeitung von SQL-Abfragen auf einer und mehrerer Tabelle zur Beantwortung der Fragestellungen (JOIN, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN,</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• <b>erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A),</b></li> <li>• <b>analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A),</b></li> <li>• <b>analysieren und erläutern eine Datenbankmodellierung (A),</b></li> <li>• <b>erläutern die Eigenschaften normalisierter Datenbankschemata (A),</b></li> <li>• <b>bestimmen Primär- und Sekundärschlüssel (M),</b></li> <li>• <b>ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M),</b></li> </ul>	<p><i>Mögliche Beispiele:</i> Schülerbefragung, Schulbuchausleihe, VideoCenter</p>

<p>SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, &lt;&gt;, &gt;, &lt;, &gt;=, &lt;=, LIKE, BETWEEN, IN, IS NULL)</p> <p>(c) Vertiefung an einem weiteren Datenbankbeispiel</p>	<ul style="list-style-type: none"> <li>• <b>modifizieren eine Datenbankmodellierung (M),</b></li> <li>• <b>modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M),</b></li> <li>• <b>überführen Datenbankschemata in vorgegebene Normalformen (M),</b></li> </ul>	
<p><b>2. Modellierung von relationalen Datenbanken</b></p> <p>(a) Entity-Relationship-Diagramm</p> <ul style="list-style-type: none"> <li>• Ermittlung von Entitäten, zugehörigen Attributen, Relationen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms</li> <li>• Erläuterung und Modifizierung einer Datenbankmodellierung</li> </ul> <p>(b) Entwicklung einer Datenbank aus einem Datenbankentwurf</p> <ul style="list-style-type: none"> <li>• Modellierung eines relationalen Datenbankschemas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln</li> </ul> <p>(c) Redundanz, Konsistenz und Normalformen</p> <ul style="list-style-type: none"> <li>• Untersuchung einer Datenbank hinsichtlich Konsistenz und Redundanz in einer Anwendungssituation</li> <li>• Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten)</li> </ul>	<ul style="list-style-type: none"> <li>• <b>verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I),</b></li> <li>• <b>ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D),</b></li> <li>• <b>stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D),</b></li> <li>• <b>überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D),</b></li> <li>• <b>implementieren ein relationales Datenbankschema als Datenbank (I),</b></li> <li>• <b>implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),</b></li> <li>• <b>untersuchen und bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen, die Sicherheit von Informatiksystemen sowie die Einhaltung der Datenschutzbestimmungen und des Urheberrechts (A),</b></li> </ul>	<p><i>Mögliche Beispiele:</i> Schulverwaltung, Buchungssystem, Fahrradverleih, Online-Kursheft</p>

<p><b>3. Datenbankbindung in Java</b></p> <p>(a) Umsetzung einer Datenbank in einem Datenbankmanagementsystem (falls in 1. oder 2. noch nicht geschehen)</p> <p>(b) Implementierung der Kontrollklassen mit Anbindung an eine Datenbank unter Verwendung didaktischer Klassen</p> <p>(c) Integration in eine GUI</p>	<ul style="list-style-type: none"> <li>• <b>untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A).</b></li> </ul>	<p><i>Mögliche Beispiele:</i> <i>Online-Kursheft</i></p>
<p><b>4. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht</b></p>		<p><i>Mögliche Beispiele:</i> <i>Online-Kursheft</i></p> <p><i>Evtl. Planspiel Datenschutz 2.0</i></p>

## Unterrichtsvorhaben Q2-III:

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

**Leitfragen:** Was sind die strukturellen Hauptbestandteile eines Computers und wie kann man sich die Ausführung eines maschinenahen Programms mit diesen Komponenten vorstellen?

**Zeitbedarf:** 6 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p><b>1. Von-Neumann-Architektur und die Ausführung maschinennaher Programme</b></p> <p>a) prinzipieller Aufbau einer von Neumann-Architektur mit CPU, Rechenwerk, Steuerwerk, Register und Hauptspeicher</p> <p>b) einige maschinennahe Befehle und ihre Repräsentation in einem Binär-Code, der in einem Register gespeichert werden kann</p> <p>c) Analyse und Erläuterung der Funktionsweise eines einfachen maschinennahen Programms</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"><li>• <b>erläutern die Ausführung eines einfachen maschinennahen Programms sowie die Datenspeicherung auf einer „Von-Neumann-Architektur“ (A).</b></li></ul>	<p><i>Beispiel:</i> Rollenspiel von Neumann-Architektur aus <a href="http://informatik-erleben.aau.at">http://informatik-erleben.aau.at</a> (E-H1 – E-H3) Addition von 4 zu einer eingegebenen Zahl mit einem Rechnermodell</p>

## Unterrichtsvorhaben Q2-IV:

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

---

### III) Qualifikationsphase – LEISTUNGSKURS

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

Die Schülerinnen und Schüler

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



## Unterrichtsvorhaben Q1-I:

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

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

**Zeitbedarf:** 15 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><b>Wiederholung und Erweiterung der objektorientierten Modellierung und Programmierung durch Analyse und Erweiterung eines kontextbezogenen Beispiels</b></p> <ul style="list-style-type: none"> <li>(a) Analyse der Problemstellung</li> <li>(b) Analyse der Modellierung (Implementationsdiagramm)</li> <li>(c) Erweiterung der Modellierung im Implementationsdiagramm (Vererbung, abstrakte Klasse, Interface)</li> <li>(d) Kommunikation zwischen mindestens zwei Objekten (grafische Darstellung)</li> <li>(e) Dokumentation von Klassen</li> <li>(f) Implementierung der Anwendung oder von Teilen der Anwendung</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• <b>analysieren und erläutern objektorientierte Modellierungen (A),</b></li> <li>• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li> <li>• <b>modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M),</b></li> <li>• <b>ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M),</b></li> <li>• entwickeln mit didaktisch orientierten Entwicklungsumgebungen einfache Benutzungsoberflächen zur Kommunikation mit einem Informatiksys-</li> </ul>	<p><i>Beispiel für eine Vorgehensweise:</i> Medien-Datenbank</p> <ul style="list-style-type: none"> <li>- Vertiefung Grundprinzipien OOM</li> <li>- Dokumentation von Quellcode</li> <li>- Objekt- &amp;, Implementationsdiagramme</li> <li>- Casten, Subtyping, Polymorphie</li> <li>- Interface, Abstrakte Klassen und Methoden am Bsp. der Klasse Medium</li> </ul> <p><i>Medien:</i> Java-Editor</p> <p>Alternative Beispiele: Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.I <a href="#">(Download LK-Q1-I.1)</a> <a href="#">(Download LK-Q1-I.2)</a></p>

---

	<p>tem (M),</p> <ul style="list-style-type: none"><li>• <b>modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),</b></li><li>• <b>implementieren Klassen in einer Programmiersprache</b> auch unter Nutzung dokumentierter Klassenbibliotheken (I),</li><li>• nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li><li>• wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I),</li><li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li><li>• testen Programme systematisch anhand von Beispielen (I),</li><li>• <b>stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D),</b></li><li>• <b>dokumentieren Klassen. (D)</b></li></ul>	
--	--	--

## Unterrichtsvorhaben Q1-II:

**Thema:** Modellierung und Implementierung von dynamischen linearen Datenstrukturen und von Anwendungen mit dynamischen linearen Datenstrukturen in kontextbezogenen Problemstellungen

**Leitfrage:** *Wie kann eine uns vorher unbekannte Anzahl an linear angeordneten Daten verwaltet werden?*

**Zeitbedarf:** 25 Stunden

### Sequenzierung des Unterrichtsvorhabens:

*Die Sequenzen 1, 2 und 3 können auch in anderer Reihenfolge unterrichtet werden.*

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><b>1. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse <code>Queue</code></b></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse <code>Queue</code></p> <p>(c) Implementierung der Klasse <code>Queue</code></p> <p>(d) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse <code>Queue</code></p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• <b>modellieren Klassen mit ihren Attributen, Methoden und Assoziationsbeziehungen unter Angabe von Multiplizitäten (M),</b></li> <li>• <b>erläutern Operationen dynamischer (linearer) Datenstrukturen (A),</b></li> <li>• analysieren und erläutern Algorithmen und Programme (A),</li> <li>• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li> <li>• <b>ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare Datensammlungen zu (M),</b></li> <li>• ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften,</li> </ul>	<p><i>Beispiel:</i> Patientenwarteschlange</p> <p>Alternative Beispiele: Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.II <a href="#">(Download LK-Q1-II.1)</a> <a href="#">(Download LK-Q1-II.2)</a></p>
<p><b>2. Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse <code>Stack</code></b></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der</p>	<ul style="list-style-type: none"> <li>• <b>ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare Datensammlungen zu (M),</b></li> <li>• ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften,</li> </ul>	<p><i>Beispiel:</i> Palindrom</p> <p>Alternative <i>Beispiel:</i> (Stapel) Handdechiffrierer mit Undo-Operation. In einem Text sind Buchstaben vertauscht worden, er wird so zu einem Geheimtext. Dieser Text wird in ein Softwareprodukt kopiert und</p>

<p>Klasse Stack</p> <p>(c) Implementierung der Klasse Stack</p> <p>(d) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Stack</p>	<p>ihre Operationen und ihre Beziehungen (M),</p> <ul style="list-style-type: none"> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),</li> <li>• modifizieren Algorithmen und Programme (I),</li> <li>• <b>implementieren Operationen dynamischer (linearer) Datenstrukturen (I),</b></li> <li>• <b>implementieren iterative auch unter Verwendung von dynamischen Datenstrukturen (I),</b></li> <li>• nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>• testen Programme systematisch anhand von Beispielen (I),</li> <li>• <b>stellen lineare Strukturen grafisch dar und erläutern ihren Aufbau (D).</b></li> </ul>	<p>schrittweise dechiffriert. Es können für jedes Zeichen Ersetzungsvorschläge gemacht werden und die Software setzt diese Ersetzung im gesamten Text um. Am Kontrolltext bemerkt man, ob die Ersetzungen sinnvoll waren. Falls man einen Irrweg gegangen ist, kann man die Ersetzungen bis zu dem gewünschten Punkt schrittweise rückgängig machen. Die Modellierung und die Implementation erfolgt mit dem generischen Datentyp <code>Stack</code>.</p> <p><i>Alternative Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.II (<a href="#">Download LK-Q1-II.3</a>)</p>
<p><b>3. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse <code>List</code></b></p> <p>(a) Erarbeitung der Vorteile der Klasse <code>List</code> im Gegensatz zu den bereits bekannten linearen Strukturen</p> <p>(b) Implementierung der Klasse lineare Liste.</p> <p>(c) Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse <code>List</code>.</p>	<p>ihre Operationen und ihre Beziehungen (M),</p> <ul style="list-style-type: none"> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),</li> <li>• modifizieren Algorithmen und Programme (I),</li> <li>• <b>implementieren Operationen dynamischer (linearer) Datenstrukturen (I),</b></li> <li>• <b>implementieren iterative auch unter Verwendung von dynamischen Datenstrukturen (I),</b></li> <li>• nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>• testen Programme systematisch anhand von Beispielen (I),</li> <li>• <b>stellen lineare Strukturen grafisch dar und erläutern ihren Aufbau (D).</b></li> </ul>	<p><i>Beispiel:</i> Einkaufsliste</p> <p><i>Alternatives Beispiel:</i> (Liste) Todo-Liste Es werden Aufgaben verwaltet, die in eine vom Benutzer gewünschte Reihenfolge angeordnet werden. Der Benutzer kann in der Todo-Liste navigieren, bestehende Einträge ändern oder löschen, sowie vor oder hinter einem Eintrag neue Aufgaben einfügen. Ein Test, in dem alle Aufgaben aufgelistet sind, kann ebenfalls abgerufen werden. Die Modellierung und die Implementation erfolgt mit dem generischen Datentyp <code>List</code>.</p> <p><i>Alternative Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.II (<a href="#">Download LK-Q1-II.4</a>)</p>

## Unterrichtsvorhaben Q1-III:

**Thema:** Suchen und Sortieren auf linearen Datenstrukturen sowie Beurteilung der Verfahren hinsichtlich ihrer Komplexitätsklassen

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

**Zeitbedarf:** 20 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><b>1. Sortieren in Listen und Arrays - Entwicklung bzw. Implementierung von Sortierverfahren</b></p> <p>(a) Entwicklung und Implementierung eines einfachen Sortierverfahrens für ein Feld auch unter der Verwendung von Struktogrammen</p> <p>(b) Anwendung verschiedener Sortierverfahren und Veranschaulichung an Zahlenbeispielen</p> <p>(c) Grafische Veranschaulichung der Sortierverfahren</p> <p>(d) Implementierung eines einfachen Sortierverfahrens für eine Liste</p> <p>(e) Entwicklung eines rekursiven Sortierverfahrens für ein Feld (z.B. Sortieren durch Mischen oder Quicksort)</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• analysieren und erläutern Algorithmen und Programme (A),</li> <li>• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li> <li>• <b>beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A),</b></li> <li>• <b>entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M),</b></li> <li>• modifizieren Algorithmen und Program-</li> </ul>	<p><i>Beispiel:</i> Karteiverwaltung, Skifahrer Algorithmen z.B.:</p> <ul style="list-style-type: none"> <li>- Bubblesort</li> <li>- Sortieren durch direktes Einfügen (Insertion Sort)</li> <li>- Sortieren durch direktes Auswählen (Selection Sort)</li> <li>- Quicksort</li> </ul>

<p><b>2. Rekursion</b></p> <p>(a) Das Prinzip der Rekursion an Beispielen aus anderen Fachbereichen (Mathematik, Deutsch, Kunst)</p> <p>(b) Anwenden und Darstellen eines rekursiven Funktionsaufrufs</p>	<p>me (I),</p> <ul style="list-style-type: none"> <li>• <b>implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),</b></li> <li>• <b>implementieren und erläutern iterative und rekursive Such- und Sortierverfahren unterschiedlicher Komplexitätsklassen (Speicherbedarf und Laufzeitverhalten (I),</b></li> </ul>	<p><i>Fibonacci Zahlen, Fakultät, Bild im Bild, Deutsche Sprache</i></p>
<p><b>3. Untersuchung der Effizienz der verschiedenen Sortierverfahren auf linearen Listen (z.B. „Sortieren durch direktes Einfügen“ und „Quicksort“)</b></p> <p>(a) Untersuchung der Anzahl der Vergleichsoperationen und des Speicherbedarf bei den Sortierverfahren</p> <p>(b) Beurteilung der Effizienz der Sortierverfahren</p> <p>(c) Erarbeitung von Komplexitätsklassen</p>	<ul style="list-style-type: none"> <li>• nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>• testen Programme systematisch anhand von Beispielen (I),</li> <li>• <b>stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch (z.B. in Form eines Struktogrammes) dar (D).</b></li> </ul>	<p>Effizienz von Algorithmen (Komplexitätsklassen)</p>
<p><b>4. Suchen von Daten in Listen und Arrays</b></p> <p>(a) Lineare Suche in Listen und in Arrays</p> <p>(b) Binäre Suche in Arrays als Beispiel für rekursives Problemlösen</p> <p>(c) Untersuchung der beiden Suchverfahren hinsichtlich ihrer Effizienz (Laufzeitverhalten, Speicherbedarf)</p>		<p>Iterative und rekursive Implementierung der binären Suche</p>

## Unterrichtsvorhaben Q1-IV:

**Thema:** Modellierung und Implementierung von dynamischen **nicht**-linearen Datenstrukturen und von Anwendungen mit dynamischen **nicht**-linearen Datenstrukturen in kontextbezogenen Problemstellungen.

**Leitfragen:** *Wie können Daten im Anwendungskontext mit Hilfe von Graphen und binärer Baumstrukturen verwaltet werden? Wie kann in einem Graphen ein beliebiger, alle oder der kürzeste Weg zwischen zwei Knoten effizient gefunden werden? Wie kann ich mit einem Binärbaum Daten sortiert verwalten?*

**Zeitbedarf:** 40 Stunden

### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<b>1. Analyse von Baumstrukturen in verschiedenen Kontexten</b> (a) Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit) (b) Aufbau und Darstellung von binären Bäumen	Die Schülerinnen und Schüler <ul style="list-style-type: none"> <li>• <b>erläutern Operationen dynamischer (nichtlinearer) Datenstrukturen (A),</b></li> <li>• analysieren und erläutern Algorithmen und Programme (A),</li> <li>• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li> <li>• beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A),</li> <li>• ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li> </ul>	<i>Beispiel: Stammbaum</i>
<b>2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse <code>BinaryTree</code></b> (a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext (b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms		<i>Beispiel: Morsebaum</i> <ul style="list-style-type: none"> <li>- Binärbaum am Bsp. Morsebaum (Codierung): rekursiver Aufbau</li> <li>- Decodierung</li> <li>- Traversierungsalgorithmen pre-, post-, inorder</li> <li>- Codierung</li> </ul>

<p>(c) Definition eines Binärbaums</p> <p>(d) Erarbeitung der Klasse <code>BinaryTree</code> und beispielhafte Anwendung der Operationen</p> <p>(e) Implementierung der Anwendung oder von Teilen der Anwendung</p> <p>(f) Implementierung der Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf</p>	<ul style="list-style-type: none"> <li>• <b>ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie nicht-lineare Datensammlungen zu (M),</b></li> <li>• modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),</li> <li>• verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M),</li> <li>• <b>entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M),</b></li> <li>• entwickeln mit didaktisch orientierten Entwicklungsumgebungen einfache Benutzungsoberflächen zur Kommunikation mit einem Informatiksystem (M),</li> <li>• implementieren Operationen dynamischer (linearer oder nichtlinearer) Datenstrukturen (I),</li> <li>• <b>implementieren und erläutern iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),</b></li> <li>• modifizieren Algorithmen und Programme (I),</li> <li>• nutzen die Syntax und Semantik einer</li> </ul>	
<p><b>3. Erarbeitung, Implementierung und Verwendung der Datenstruktur binärer Suchbaum im Anwendungskontext</b></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramm, grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften</p> <p>(c) Erarbeitung der Klasse <code>BinarySearchTree</code> und Einführung des Interface <code>Item</code> zur Realisierung einer geeigneten Ordnungsrelation</p> <p>(d) Implementierung des Konstruktors und der Methode <code>search</code> der Klasse <code>BinarySearchTree&lt;ContentType&gt;</code></p> <p>(e) Implementierung der Anwendung oder von Teilen der Anwendung inklusive</p>		<p><i>Beispiel:</i> Informatikerbaum als binärer Suchbaum</p> <p><i>Beispiel:</i> Huffman-Codierung (Zeichenhäufigkeit) zur Herleitung des Interface <code>Item</code></p>



einer sortierten Ausgabe des Baums		
<p><b>4. Analyse von Graphen in verschiedenen Kontexten</b></p> <p>(a) Grundlegende Begriffe (Graph, gerichtet – ungerichtet, Knoten, Kanten, Kantengewicht)</p> <p>(b) Aufbau und Darstellung von Graphen anhand von Graphenstrukturen in verschiedenen Kontexten (Adjazenzmatrix, Adjazenzliste)</p>	<p>Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</p> <ul style="list-style-type: none"> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>• testen Programme systematisch anhand von Beispielen (I),</li> <li>• <b>stellen nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D),</b></li> <li>• <b>stellen iterative und rekursive Algorithmen umgangssprachlich dar (D).</b></li> </ul>	<p><i>Beispiel:</i> Soziale Netzwerke, Wegsuche</p>
<p><b>5. Die Datenstruktur Graph im Anwendungskontext unter Nutzung der Klassen Graph, Vertex und Edge.</b></p> <p>(a) Erarbeitung der Klassen Graph, Vertex und Edge und beispielhafte Anwendung der Operationen</p> <p>(b) Bestimmung von Wegen in Graphen im Anwendungskontext (Tiefensuche, Breitensuche) und Implementierung der Algorithmen</p> <p>(c) Bestimmung von kürzesten Wegen in Graphen im Anwendungskontext (Backtracking, Dijkstra) und Implementierung der Algorithmen</p> <p>(d) Bestimmung von minimalen Spannbäumen eines Graphen im Anwendungskontext.</p>		

## Unterrichtsvorhaben Q1-V:

**Thema:** Grundlagen der Netzwerkkommunikation sowie Modellierung und Implementierung von Client-Server-Anwendungen in kontextbezogenen Problemstellungen sowie Sicherheit in Informatiksystemen.

**Leitfragen:** *Wie werden Daten in Netzen übermittelt? Wie entwickelt man ein Client-Server-System im Anwendungskontext? Welche Algorithmen sind zu implementieren? Was sollte man in Bezug auf die Sicherheit beachten?*

**Zeitbedarf:** 24 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<b>1. Sicherheit und Kryptologie</b> (a) Vertraulichkeit, Integrität, Authentizität in Netzwerken (b) symmetrische und asymmetrische kryptografische Verfahren (Cäsar-, Vigenère-, RSA-Verfahren) als Methoden Daten im Netz verschlüsselt zu übertragen (c) Implementierung symmetrischer Verschlüsselungsverfahren	Die Schülerinnen und Schüler <ul style="list-style-type: none"> <li>• analysieren und erläutern Protokolle zur Kommunikation in einem Client-Server-Netzwerk (A),</li> <li>• beschreiben und erläutern Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A),</li> <li>• analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A),</li> <li>• analysieren und erläutern Algorithmen und Programme (A),</li> <li>• untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen (A),</li> </ul>	<i>Material: Spion-Camp</i>  <i>Kryptologie</i> <i>am Bsp. einer Geburtstagsfeier</i> <ul style="list-style-type: none"> <li>• Caesar-Verfahren</li> <li>• Vigenère-Verfahren</li> <li>• RSA</li> </ul> <i>Diffie-Hellmann-Schlüsselaustauschverfahren</i>
<b>2. Grundlagen den Datenübertragung in Netzwerken</b> (a) <i>Schichtenmodell</i> Erarbeitung eines standardisierten Schichtenmodells für Netzwerkkommunikation (b) <i>Grundlagen der Codierung</i> Erarbeitung einer eigenen,	(siehe oben)	<i>Beispiel: Kommunikation übers Netz (Gold-city etc.)</i>  <i>Beispiel: Schulnetz und LAN-Party</i> <ul style="list-style-type: none"> <li>• Datenübertragung (Paketübermittlung)</li> <li>• Netzwerktopologien</li> <li>• Protokolle</li> </ul>

<p>vereinfachten Codierung für die Bitübertragung</p> <p>(c) <i>Topologien</i> Erarbeitung und Vergleich ausgewählter Netzwerktopologien</p> <p>(d) <i>Routing</i> Analyse von Grundlagen der Adressierung in IP-Netzwerken</p> <p>(e) <i>Protokolle</i> Erarbeitung des beispielhaften Aufbaus eines Protokoll auf der Anwendungsschicht</p>	<ul style="list-style-type: none"> <li>• ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihr Operationen und Beziehungen (M),</li> <li>• modellieren Klassen mit ihren Attributen, Methoden und Assoziationsbeziehungen unter Angabe von Multiplizitäten (M),</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare und nichtlineare Datensammlungen zu (M),</li> <li>• analysieren und erläutern objektorientierte Modellierungen (A),</li> <li>• stellen Klassen und ihre Beziehungen grafisch dar (D),</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),</li> <li>• nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D).</li> </ul>	<ul style="list-style-type: none"> <li>• <i>TCP/IP-Schichtenmodell</i></li> </ul> <p><i>Materialien:</i> <i>Filius, Film „Sendung mit der Maus“</i></p>
<p><b>3. Analyse, Modellierung und Implementierung von Netzerkennungen in Client-Server-Struktur</b></p> <p>(a) <i>Nutzung einfacher Server-Dienste mittels Client</i> Modellierung und Implementierung von Clients für einfach Serverdienste</p> <p>(b) <i>Anbieten von Diensten mittels Server</i></p> <ul style="list-style-type: none"> <li>• Analyse vorgegebener Implementationen einfacher Server</li> <li>• Modellierung und Implementierung eigener Server</li> </ul>		<p><i>Beispiel:</i></p> <ul style="list-style-type: none"> <li>• Echo- bzw. Daytime-Clients und – Server sowie eigene Erweiterungen</li> <li>• Pop3-Client</li> </ul>
<p><b>4. Entwicklung eines vollständigen Client-Server-Systems</b></p> <ul style="list-style-type: none"> <li>• Protokollentwurf</li> <li>• Modellierung mittels Entwurfs- und Implementationsdiagramm</li> <li>• Bedeutung von Nebenläufigkeit</li> <li>• Implementierung</li> </ul>		<p><i>Beispiel:</i> Messenger-Dienst (Chat)</p>

## Unterrichtsvorhaben Q2-I:

**Thema:** Endliche Automaten und formale Sprachen

**Leitfragen:** *Wie kann man (endliche) Automaten genau beschreiben? Wie können endliche Automaten und Kellerautomaten (in alltäglichen Kontexten oder zu informatischen Problemstellungen) modelliert werden? Wie können Sprachen durch Grammatiken beschrieben werden? Welche Zusammenhänge gibt es zwischen formalen Sprachen, Automaten und Grammatiken? Wie kann ein Scanner, ein Parser und ein Interpreter für eine einfache formale Sprache entwickelt werden? Welche Möglichkeiten bieten Informatiksysteme und wo liegen ihre Grenzen?*

**Zeitbedarf:** 27 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<b>1. Endliche Automaten</b> (a) Von realen Automaten zur formalen Beschreibung eines endlichen Automaten (b) Untersuchung, Darstellung und Entwicklung von Transduktoren, Akzeptoren und nichtdeterministischen endlichen Automaten (c) Umwandlung nichtdeterministischer endlicher Automaten in deterministische endliche Automaten	Die Schülerinnen und Schüler <ul style="list-style-type: none"> <li>• <b>analysieren und erläutern die Eigenschaften endlicher Automaten und Kellerautomaten einschließlich ihres Verhaltens auf bestimmte Eingaben (A),</b></li> <li>• <b>analysieren und erläutern Grammatiken regulärer und kontextfreier Sprachen (A),</b></li> <li>• <b>erläutern die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang (A),</b></li> <li>• <b>ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A),</b></li> </ul>	<i>Beispiele:</i> Ticketautomat der Bahn, Cola-Automat, Geldspielautomat, Akzeptor für bestimmte Zahlungsmengen
<b>2. Untersuchung und Entwicklung von Grammatiken regulärer Sprachen</b> (a) Erarbeitung der formalen Darstellung regulärer Grammatiken	(siehe oben)	<i>Beispiele:</i> Die deutsche Grammatik (Satzgliederungsgrammatik), reguläre Grammatik für Wörter,

<p>(b) Untersuchung, Modifikation und Entwicklung von Grammatiken</p> <p>(c) Entwicklung von endlichen Automaten zum Erkennen regulärer Sprachen die durch Grammatiken gegeben werden</p> <p>(d) Entwicklung regulärer Grammatiken zu endlichen Automaten</p> <p>(e) Entwicklung eines Scanners, Parsers und Interpreters für eine einfache reguläre Sprache</p>	<ul style="list-style-type: none"> <li>• <b>entwickeln und modifizieren zu einer Problemstellung endliche Automaten oder Kellerautomaten (M),</b></li> <li>• <b>entwickeln zur akzeptierten Sprache eines Automaten die zugehörige Grammatik (M),</b></li> <li>• <b>entwickeln zur Grammatik einer regulären oder kontextfreien Sprache einen zugehörigen endlichen Automaten oder einen Kellerautomaten (M),</b></li> <li>• <b>modifizieren Grammatiken regulärer und kontextfreier Sprachen (M),</b></li> <li>• <b>entwickeln zu einer regulären oder kontextfreien Sprache eine Grammatik, die die Sprache erzeugt (M),</b></li> <li>• <b>stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D),</b></li> <li>• <b>ermitteln die Sprache, die ein endlicher Automat oder ein Kellerautomat akzeptiert (D).</b></li> <li>• <b>beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D),</b></li> <li>• <b>modellieren und implementieren Scanner, Parser und Interpreter zu einer gegebenen regulären Sprache (I),</b></li> <li>• <b>implementieren iterative oder rekursive Algorithmen auch unter Verwendung dynamischer Datenstrukt-</b></li> </ul>	<p>die bestimmte Zahlen repräsentieren</p>
<p><b>3. Grenzen endlicher Automaten</b></p> <p>a) Erarbeitung von Grenzen endlicher Automaten</p> <p>b) Als möglicher Ausweg: Entwicklung eines Kellerautomaten</p> <p>c) Anwendung eines Kellerautomaten zur Syntaxüberprüfung auf Grundlage von nicht-regulären Grammatiken</p> <p>d) Implementierung eines Kellerautomaten zur Syntaxüberprüfung</p>	<ul style="list-style-type: none"> <li>• <b>entwickeln und modifizieren zu einer Problemstellung endliche Automaten oder Kellerautomaten (M),</b></li> <li>• <b>entwickeln zur akzeptierten Sprache eines Automaten die zugehörige Grammatik (M),</b></li> <li>• <b>entwickeln zur Grammatik einer regulären oder kontextfreien Sprache einen zugehörigen endlichen Automaten oder einen Kellerautomaten (M),</b></li> <li>• <b>modifizieren Grammatiken regulärer und kontextfreier Sprachen (M),</b></li> <li>• <b>entwickeln zu einer regulären oder kontextfreien Sprache eine Grammatik, die die Sprache erzeugt (M),</b></li> <li>• <b>stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D),</b></li> <li>• <b>ermitteln die Sprache, die ein endlicher Automat oder ein Kellerautomat akzeptiert (D).</b></li> <li>• <b>beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D),</b></li> <li>• <b>modellieren und implementieren Scanner, Parser und Interpreter zu einer gegebenen regulären Sprache (I),</b></li> <li>• <b>implementieren iterative oder rekursive Algorithmen auch unter Verwendung dynamischer Datenstrukt-</b></li> </ul>	<p><i>Beispiele:</i> Klammerausdrücke, <math>a^n b^n</math> im Vergleich zu <math>(ab)^n</math></p>
<p><b>4. Grenzen der Automatisierbarkeit</b></p> <p>a) Untersuchung verschiedener nicht in polynomialer Zeit berechenbarer Probleme der Informatik</p> <p>b) Vorstellung des Halteproblems</p> <p>c) Unlösbarkeit des Halteproblems</p> <p>d) Beurteilung des Einsatzes von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzi-</p>	<ul style="list-style-type: none"> <li>• <b>entwickeln und modifizieren zu einer Problemstellung endliche Automaten oder Kellerautomaten (M),</b></li> <li>• <b>entwickeln zur akzeptierten Sprache eines Automaten die zugehörige Grammatik (M),</b></li> <li>• <b>entwickeln zur Grammatik einer regulären oder kontextfreien Sprache einen zugehörigen endlichen Automaten oder einen Kellerautomaten (M),</b></li> <li>• <b>modifizieren Grammatiken regulärer und kontextfreier Sprachen (M),</b></li> <li>• <b>entwickeln zu einer regulären oder kontextfreien Sprache eine Grammatik, die die Sprache erzeugt (M),</b></li> <li>• <b>stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D),</b></li> <li>• <b>ermitteln die Sprache, die ein endlicher Automat oder ein Kellerautomat akzeptiert (D).</b></li> <li>• <b>beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D),</b></li> <li>• <b>modellieren und implementieren Scanner, Parser und Interpreter zu einer gegebenen regulären Sprache (I),</b></li> <li>• <b>implementieren iterative oder rekursive Algorithmen auch unter Verwendung dynamischer Datenstrukt-</b></li> </ul>	<p><i>Beispiel:</i> Travelling Salesman Problem usw., Halteproblem, Affenpuzzle</p>

---

pieller Grenzen	<b>ren (I),</b> <ul style="list-style-type: none"><li>• <b>untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A),</b></li><li>• nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D).</li></ul>	
-----------------	--	--

## Unterrichtsvorhaben Q2-II:

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

**Leitfragen:** Was sind die strukturellen Hauptbestandteile eines Computers und wie kann man sich die Ausführung eines maschinennahen Programms mit diesen Komponenten vorstellen?

**Zeitbedarf:** 10 Stunden

### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<b>1. Von-Neumann-Architektur und die Ausführung maschinennaher Programme</b> a) prinzipieller Aufbau einer von Neumann-Architektur mit CPU, Rechenwerk, Steuerwerk, Register und Hauptspeicher b) einige maschinennahe Befehle und ihre Repräsentation in einem Binär-Code, der in einem Register gespeichert werden kann c) Analyse und Erläuterung der Funktionsweise eines einfachen maschinennahen Programms	Die Schülerinnen und Schüler <ul style="list-style-type: none"><li>• erläutern die Ausführung eines einfachen maschinennahen Programms sowie die Datenspeicherung auf einer „Von-Neumann-Architektur“ (A),</li><li>• erläutern die Funktionsweise eines Compilers (A).</li></ul>	<i>Beispiel:</i> Rollenspiel von Neumann-Architektur aus <a href="http://informatik-erleben.aau.at">http://informatik-erleben.aau.at</a> (E-H1 – E-H3) Addition von 4 zu einer eingegebenen Zahl mit einem Rechnermodell
<b>2. Untersuchung der Phasen eines Compilers</b> a) Prozesse beim Compiler: Scannen, parsen, übersetzen/interpretieren		<i>Beispiel:</i> Scanner, Parser und Interpreter einer Konstruktionsprache für geometrische Figuren

---

<p>b) Arten der Fehler: lexikalischer, syntaktischer, semantischer Fehler</p> <p>c) Analyse und Erläuterung der Funktionsweise eines einfachen maschinennahen Programms</p> <p>d) Einordnung der neuen Begriffe in den Gesamtkontext der formalen Sprachen (Rückgriff auf UV Q2-I)</p>		
--	--	--



## Unterrichtsvorhaben Q2-III:

**Thema:** Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten und Sicherheit und Datenschutz in Informatiksystemen

**Leitfragen:** *Wie können Fragestellungen mit Hilfe einer Datenbank beantwortet werden? Wie entwickelt man selbst eine Datenbank für einen Anwendungskontext? Welche datenschutzrechtlichen Vorgaben sind zu beachten? Welche moralische und rechtliche Verantwortung tragen Informatikerinnen und Informatiker hinsichtlich des Datenschutzes, des Urheberrechts und der gesellschaftlichen Auswirkungen informatischer Systeme?*

**Zeitbedarf:** 40 Stunden

### Sequenzierung des Unterrichtsvorhabens

Anmerkung: Unterrichtssequenz 1 und 2 können in ihrer Reihenfolge auch getauscht werden und greifen in komplexen Beispielen ineinander.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<b>1. Nutzung von relationalen Datenbanken</b> (a) Aufbau von Datenbanken und Grundbegriffe <ul style="list-style-type: none"><li>• Entwicklung von Fragestellungen zur vorhandenen Datenbank</li><li>• Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, Datenbankschema</li></ul> (b) SQL-Abfragen <ul style="list-style-type: none"><li>• Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL (<code>SELECT (DISTINCT) ...FROM, WHERE, AND, OR, NOT</code>) auf einer Tabelle</li></ul>	Die Schülerinnen und Schüler <ul style="list-style-type: none"><li>• <b>erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A),</b></li><li>• <b>analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A),</b></li><li>• <b>analysieren und erläutern eine Datenbankmodellierung (A),</b></li><li>• <b>erläutern die Eigenschaften normalisierter Datenbankschemata (A),</b></li><li>• <b>bestimmen Primär- und Sekundärschlüssel (M),</b></li></ul>	<i>Mögliche Beispiele:</i> Schülerbefragung, Schulbuchausleihe, VideoCenter

<ul style="list-style-type: none"> <li>• Analyse und Erarbeitung von SQL-Abfragen auf einer und mehrerer Tabellen zur Beantwortung der Fragestellungen (JOIN, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, &lt;&gt;, &gt;, &lt;, &gt;=, &lt;=, LIKE, BETWEEN, IN, IS NULL)</li> </ul> <p>(c) Vertiefung an einem weiteren Datenbankbeispiel</p>	<ul style="list-style-type: none"> <li>• <b>ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M),</b></li> <li>• <b>modifizieren eine Datenbankmodellierung (M),</b></li> <li>• <b>modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M),</b></li> <li>• <b>überführen Datenbankschemata in vorgegebene Normalformen (M),</b></li> <li>• <b>verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I),</b></li> <li>• <b>ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D),</b></li> <li>• <b>stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D),</b></li> <li>• <b>überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D),</b></li> <li>• <b>implementieren ein relationales Datenbankschema als Datenbank (I),</b></li> <li>• <b>implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),</b></li> <li>• <b>untersuchen und bewerten anhand</b></li> </ul>	
<p><b>2. Modellierung von relationalen Datenbanken</b></p> <p>(a) Entity-Relationship-Diagramm</p> <ul style="list-style-type: none"> <li>• Ermittlung von Entitäten, zugehörigen Attributen, Relationen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms</li> <li>• Erläuterung und Modifizierung einer Datenbankmodellierung</li> </ul> <p>(b) Entwicklung einer Datenbank aus einem Datenbankentwurf</p> <ul style="list-style-type: none"> <li>• Modellierung eines relationalen Datenbankschemas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln</li> </ul> <p>(c) Redundanz, Konsistenz und Normalformen</p> <ul style="list-style-type: none"> <li>• Untersuchung einer Datenbank hinsichtlich Konsistenz und Redundanz in</li> </ul>	<ul style="list-style-type: none"> <li>• <b>ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M),</b></li> <li>• <b>modifizieren eine Datenbankmodellierung (M),</b></li> <li>• <b>modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M),</b></li> <li>• <b>überführen Datenbankschemata in vorgegebene Normalformen (M),</b></li> <li>• <b>verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I),</b></li> <li>• <b>ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D),</b></li> <li>• <b>stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D),</b></li> <li>• <b>überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D),</b></li> <li>• <b>implementieren ein relationales Datenbankschema als Datenbank (I),</b></li> <li>• <b>implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),</b></li> <li>• <b>untersuchen und bewerten anhand</b></li> </ul>	<p><i>Mögliche Beispiele:</i> Schulverwaltung, Buchungssystem, Fahrradverleih, Online-Kursheft</p>

<p>einer Anwendungssituation</p> <ul style="list-style-type: none"> <li>• Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten)</li> </ul>	<p><b>von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen, die Sicherheit von Informatiksystemen sowie die Einhaltung der Datenschutzbestimmungen und des Urheberrechts (A),</b></p>	
<p><b>3. Datenbankanbindung in Java</b></p> <p>(a) Umsetzung einer Datenbank in einem Datenbankmanagementsystem (falls in 1. oder 2. noch nicht geschehen)</p> <p>(b) Implementierung der Kontrollklassen mit Anbindung an eine Datenbank unter Verwendung didaktischer Klassen</p> <p>(c) Integration in eine GUI</p>	<ul style="list-style-type: none"> <li>• <b>untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A),</b></li> <li>• nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D).</li> </ul>	<p><i>Mögliche Beispiele: Online-Kursheft</i></p>
<p><b>4. Fallbeispiele zur Datenschutzproblematik, zum Urheberrecht und der moralischen Verantwortung</b></p> <p>(a) Erarbeitung von Interesse verschiedener Interessengruppen im Hinblick auf Fallbeispiele</p> <p>(b) Erarbeitung von Grundideen des Datenschutzes (z.B. personenbezogene Daten, informationelles Selbstbestimmungsrecht, Datensparsamkeit usw.)</p> <p>(c) Erarbeitung von Grundideen des Urheberrechts</p> <p>(d) Erarbeitung eines einfachen moralischen Bewertungsmaßstabes</p> <p>(e) Anwendung auf Fallbeispiele</p>		<p><i>Datenschutz z.B. am Beispiel eines Online-Kursheftes</i></p> <p><i>Ethik am Beispiel der Verantwortungsethik nach Hans Jonas Erarbeitung an verschiedenen Fallbeispielen wie bspw. autonomes Fahren, Software für Drohnen, Bewertungssystem für chinesische Bürger</i></p>

---

<p>(f) Einführung in das Verfassen von Stellungnahmen sowie in Kriterien für gelungene Stellungnahmen</p> <p>(g) Erarbeitung von Stellungnahmen zu Fallbeispielen</p>		
---	--	--

## Unterrichtsvorhaben Q2-IV:

**Thema:** Umsetzung eines selbstgewählten Projekts – Durchführung eines vollständigen Softwareentwicklungszyklus

**Leitfragen:** *Wie ist ein Softwareentwicklungszyklus aufgebaut? Welche Anforderungen muss unsere Software erfüllen? Welche Rollen muss es in einem Softwareprojekt geben und wie verteilen wir die Aufgaben?*

**Zeitbedarf:** 20 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p><b>1. Analyse einer lebensweltnahen Problemstellung im Hinblick auf die Entwicklung eines Java-Programms mit Datenbankanbindung</b></p> <p>(a) Erläuterung eines Vorgehensmodells zur Softwareentwicklung</p> <p>(b) Entwicklung einer Programmidee</p> <p>(c) Analyse des Problembereichs</p> <p>(d) Entwicklung eines Anforderungskatalogs für das zu entwickelnde Programm</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"><li>• ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li><li>• modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M),</li><li>• modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),</li></ul>	<p><i>Beispiel:</i></p> <p>Netzwerkspiel, Software mit Datenbankanbindung, App-Programmierung für Android, personalisierter Vertretungsplan usw.</p>

<p><b>2. Modellierung einer Problemlösung</b></p> <ul style="list-style-type: none"> <li>(a) Modellierung der für die Umsetzung notwendigen Klassen usw. (Entwurfs- &amp; Implementationsdiagramm)</li> <li>(b) Modellierung der grafischen Benutzeroberfläche</li> <li>(c) Aufteilung der Arbeiten im Projektteam</li> <li>(d) Festlegung eines Zeitplans (Meilensteine)</li> </ul>		
<p><b>3. Umsetzung des Softwareprojekts</b></p> <ul style="list-style-type: none"> <li>(a) Implementierung der notwendigen Klassen unter Verwendung didaktischer Klassen</li> <li>(b) Integration in den Prototypen der Benutzeroberfläche</li> </ul>		
<p><b>4. Test und Evaluation der Software</b></p> <ul style="list-style-type: none"> <li>(a) Installation und Test des Endprodukts im konkreten Anwendungskontext</li> <li>(b) Beseitigung eventuell auffallender Fehler in der Testphase</li> <li>(c) Präsentation des Endprodukts vor dem "Kunden"</li> <li>(d) Analyse möglicher Erweiterungen</li> <li>(e) Ggf. Schulung von Anwendern der neuen Software</li> </ul>		
<p><b>5. Evaluation und Reflexion des Pro-</b></p>		

---

<b>jekts</b> (a) Reflexion der Projektplanung, der Arbeit im Team, der Meilensteine, der Präsentation vor dem "Kunden" (b) Identifikation der Stufen eines Softwareentwicklungszyklus		
---	--	--

### **Unterrichtsvorhaben Q2-V:**

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

---

## 2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

In Absprache mit der Lehrerkonferenz sowie unter Berücksichtigung des Schulprogramms hat die Fachkonferenz Informatik des Ravensberger Gymnasiums die folgenden fachmethodischen und fachdidaktischen Grundsätze beschlossen. In diesem Zusammenhang beziehen sich die Grundsätze 1 bis 13 auf fächerübergreifende Aspekte, die auch Gegenstand der Qualitätsanalyse sind, die Grundsätze 14 bis 20 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 Lehr- und Lernzeit wird intensiv für Unterrichtszwecke genutzt.
- 13) Es herrscht ein positives pädagogisches Klima im Unterricht.

### Fachliche Grundsätze:

- 14) Der Unterricht unterliegt der Wissenschaftsorientierung und ist dementsprechend eng verzahnt mit seiner Bezugswissenschaft.
- 15) Der Unterricht ist problemorientiert.
- 16) Der Unterricht folgt dem Prinzip der Exemplarizität und soll ermöglichen, informatische Strukturen und Gesetzmäßigkeiten in den ausgewählten Problemen und Projekten zu erkennen.
- 17) Der Unterricht ist anschaulich sowie gegenwarts- und zukunftsorientiert und gewinnt dadurch für die Schülerinnen und Schüler an Bedeutsamkeit.



- 
- 18) Der Unterricht ist handlungsorientiert, d.h. projekt- und produktorientiert angelegt.
  - 19) Im Unterricht werden sowohl für die Schule didaktisch reduzierte als auch reale Informatiksysteme aus der Wissenschafts-, Berufs- und Lebenswelt eingesetzt.
  - 20) Der Unterricht beinhaltet reale Begegnung mit Informatiksystemen.

---

## 2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

Auf der Grundlage von §13 - §16 der APO-GOST sowie Kapitel 3 des Kernlehrplans Informatik für die gymnasiale Oberstufe hat die Fachkonferenz 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
- Grundkurse Q 1.1: 2 Klausuren  
Dauer der Klausuren: 90 Minuten
- Leistungskurse Q 1.1: 2 Klausuren  
Dauer der Klausuren: 135 Minuten
- Grundkurse Q 1.2: 2 Klausuren  
Dauer der Klausuren: 135 Minuten
- Leistungskurse Q 1.2: 2 Klausuren  
Dauer der Klausuren: 180 Minuten
- Grundkurse Q 2.1: 2 Klausuren  
Dauer der Klausuren: 135 Minuten
- Leistungskurse Q 2.1: 2 Klausuren  
Dauer der Klausuren: 180 Minuten
- Grundkurse Q 2.2: 1 Klausur unter Abiturbedingungen
- Leistungskurse Q 2.2: 1 Klausur unter Abiturbedingungen
- Anstelle einer Klausur kann gemäß dem Beschluss der Lehrerkonferenz in Q 2.1 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 Hilfspunkte, 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.

#### Verbindliche Absprachen der Fachkonferenz für Leistungskurse

- Im ersten Jahr der Qualifikationsphase entwickeln und dokumentieren die Schülerinnen und Schüler im Rahmen einer Langzeithausaufgabe selbstständig eine Lösung zu einer Problemstellung, die durch die Aufgaben der ersten Runde des Bundeswettbewerbes Informatik vorgegeben wird.
- Im zweiten Jahr der Qualifikationsphase erstellen, dokumentieren und präsentieren die Schülerinnen und Schüler in Kleingruppen ein anwendungsbezogenes Softwareprodukt. Dies wird in die Note für die Sonstige Mitarbeit einbezogen.

#### Leistungsaspekte

##### Mündliche Leistungen

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

---

### Praktische Leistungen am Computer

- Implementierung, Test und Anwendung von Informatiksystemen

### Sonstige schriftliche Leistungen

- Hausaufgaben
- Dokumentationen zu einem Unterrichtsvorhaben
- Arbeitsmappe und Aufzeichnungen zu einem durchgeführten Unterrichtsvorhaben
- Lernerfolgsüberprüfung durch kurze schriftliche Übungen  
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,
- Anstrengungsbereitschaft und Konzentration auf die Arbeit,
- Umgang mit neuen Problemen, Beteiligung bei der Suche nach neuen Lösungswegen,
- Umgang mit Arbeitsaufträgen (Hausaufgaben, Unterrichtsaufgaben etc.)
- 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 Teamfähigkeit,
  - 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 Schuljahres 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 oder
- durch eine individuelle Lern-/Förderempfehlung

erfolgen.

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

---

### **3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen**

Die Fachkonferenz Informatik hat sich im Rahmen des Schulprogramms für folgende zentrale Schwerpunkte entschieden:

#### **Zusammenarbeit mit anderen Fächern**

Im Informatikunterricht werden Kompetenzen anhand informatischer Inhalte in verschiedenen Anwendungskontexten erworben, in denen Schülerinnen und Schülern aus anderen Fächern Kenntnisse mitbringen können. Diese können insbesondere bei der Auswahl und Bearbeitung von Softwareprojekten berücksichtigt werden und in einem hinsichtlich der informatischen Problemstellung angemessenem Maß in den Unterricht Eingang finden. Da im Inhaltsfeld Informatik, Mensch und Gesellschaft auch gesellschaftliche und ethische Fragen im Unterricht angesprochen werden, fließen hier auch Kenntnisse, die in den Fächern Sozialwissenschaften und Philosophie erworben wurden, ein.

#### **Vorbereitung auf die Erstellung der Facharbeit**

Im zweiten Halbjahr des ersten Jahres der Qualifikationsphase werden im Unterricht an geeigneten Stellen Hinweise zur Erstellung von Facharbeiten gegeben. Das betrifft u. a. Themenvorschläge, Hinweise zu den Anforderungen und zur Bewertung. Es wird vereinbart, dass in der Regel Facharbeiten vergeben werden, die mit der eigenständigen Entwicklung eines Produktes verbunden sind. Unter einem Produkt ist dabei aber nicht nur ein Softwareprodukt zu verstehen, sondern eine Facharbeit kann ebenso gut in einem Flyer, einer Ausstellung oder einem Elternabend münden.

#### **Exkursionen**

In jedem Jahr der Qualifikationsphase wird eine zu den Unterrichtsinhalten thematisch passende Exkursion durchgeführt. Die außerunterrichtliche Veranstaltung wird im Unterricht vor- und nachbereitet und soll von den Schülern angestoßen werden.

#### **Einbindung von externen Experten in den Fachunterricht**

Insbesondere in der Qualifikationsphase werden nach Möglichkeit externe Experten in den Fachunterricht einbezogen, die den Schülern die Aktualität der Unterrichtsinhalte verdeutlichen. Im ersten Jahr der Qualifikationsphase geschieht dies beispielsweise durch einen Experten der Fachhochschule Bielefeld, der mit den Schülern das Thema *Kryptologie* erörtert.

---

## **4 Qualitätssicherung und Evaluation**

Durch Diskussion der Aufgabenstellung von Klausuren in Fachdienstbesprechungen und eine regelmäßige Erörterung der Ergebnisse von Leistungsüberprüfungen wird ein hohes Maß an fachlicher Qualitätssicherung erreicht.

Die Fachschaft Informatik wird fortlaufend die Ergebnisse aus besuchten Fortbildungsveranstaltungen in die Unterrichtsgestaltung einfließen lassen und bei Bedarf das Curriculum daraufhin anpassen.