

**Schulinterner Lehrplan  
des Gutenberg-Gymnasiums Bergheim  
zum Kernlehrplan für die gymnasiale Oberstufe**

**Informatik** (Qualifikationsphase)

**Stand: 25.05.2016**

## Inhalt

<b>1</b>	<b>Rahmenbedingungen der fachlichen Arbeit</b>	<b>3</b>
<b>2</b>	<b>Entscheidungen zum Unterricht</b>	<b>5</b>
2.1	Unterrichtsvorhaben	5
2.1.1	Übersichtsraster Unterrichtsvorhaben	6
	1) <i>Qualifikationsphase Q1 (Grundkurs)</i>	6
	2) <i>Qualifikationsphase Q2 (Grundkurs)</i>	8
2.1.2	Konkretisierte Unterrichtsvorhaben	10
	1) <i>Qualifikationsphase Q1 (Grundkurs)</i>	10
	2) <i>Qualifikationsphase Q2 (Grundkurs)</i>	20
2.2	Grundsätze der fachmethodischen und fachdidaktischen Arbeit	27
2.3	Grundsätze der Leistungsbewertung und Leistungsrückmeldung	28
2.3.1	Beurteilungsbereich Klausuren und Facharbeiten	28
	1) <i>Klausuren</i>	28
	2) <i>Facharbeiten</i>	29
2.3.2	Beurteilungsbereich Sonstige Mitarbeit	29
2.4	Lehr- und Lernmittel	32
<b>3</b>	<b>Entscheidungen zu fach- und unterrichtsübergreifenden Fragen</b>	<b>33</b>
<b>4</b>	<b>Qualitätssicherung und Evaluation</b>	<b>34</b>

# 1 Rahmenbedingungen der fachlichen Arbeit

Das **Gutenberg-Gymnasium** liegt in Bergheim im Rhein-Erft-Kreis. Dort werden ca. 850 Schülerinnen und Schüler von ca. 70 Lehrerinnen und Lehrern unterrichtet. Die Schülerinnen und Schüler kommen aus der Stadt Bergheim, viele auch aus umliegenden kleineren Gemeinden im Kreis Bergheim. Die Unterrichtsstunden dauern 45 Minuten, wobei ein hoher Prozentsatz des erteilten Unterrichts im Doppelstundensystem erteilt wird. Die Schule verfügt über gut eingerichtete Fachräume, so wurde z.B. die IT-Ausstattung in 2014 komplett aktualisiert. Alle Unterrichtsräume sind barrierefrei erreichbar.

Die Nähe zu Köln ermöglicht **Kooperationsmöglichkeiten** mit verschiedenen Einrichtungen, wie z. B. der Universität und kulturellen Institutionen. In der Sekundarstufe I ist die Schule meist vierzünftig, in der gymnasialen Oberstufe besuchen durchschnittlich 100 Teilnehmerinnen und Teilnehmer je einen Jahrgang. In der EF absolvieren die Schülerinnen und Schüler ein 14-tägiges Betriebspraktikum, in dieser Zeit werden den Seiteneinsteigern individuelle Fördermaßnahmen in einzelnen Fächern angeboten. In der Oberstufe kooperiert das Gutenberg-Gymnasium mit dem Bergheimer Erftgymnasium und dem Bedburger Silverberg-Gymnasium. Dies ermöglicht die Einrichtung einer hohen Anzahl von Kursen, wovon vor allem der Leistungskursbereich profitiert. Insbesondere in Informatik findet ein fester Kooperationskurs mit Schülern aus Bedburg am Gutenberg-Gymnasium statt.

Das Fach Informatik wird am Gutenberg-Gymnasium ab der Jahrgangsstufe 8 im **Differenzierungsbereich** zweistündig unterrichtet und von etwa der Hälfte der Schülerinnen und Schüler besucht. In der zweijährigen Laufzeit dieser Kurse wird in altersstufengerechter Weise unter anderem im Bereich praktische Informatik auf Grundlagen der Algorithmik (*Scratch, PHP*), der Datenstrukturierung (*HTML, Access*) und der Grafikbearbeitung (*Dateiformate, GIMP*) eingegangen, im Bereich theoretische Informatik wird spielerisch in die Automatentheorie (*Kara*) eingeführt und die technische Informatik am Beispiel von Schaltwerken und Schaltnetzen behandelt. Der Unterricht erfolgt dabei in enger Verzahnung mit Inhalten der Mathematik (*Logik*) und der Physik (*Hardware*) und wird zum Teil in Form von fächerverbindenden Projekten mit der Fachschaft Kunst (*Grafikdesign*) und der Fachschaft Chemie (*Entwicklung von Lernprogrammen*) und in Kooperation mit außerschulischen Partnern wie der Rheinischen Fachhochschule Köln gestaltet.

In der Jahrgangsstufe 7 wird ein für alle verpflichtender Kurs zum Umgang mit informatischen Systemen unter der Bezeichnung Informationstechnische Grundlagen (*IKG-Kurs*) durchgeführt, der jedoch nicht unmittelbar dem Fach Informatik zuzuordnen ist. Die Einbeziehung informatischer und medienbezogener Kompetenzen in die schulische Ausbildung ab der Stufe 5 wird derzeit im Rahmen eines **Medienerziehungskonzept** unter Einbeziehung der Fachschaft Informatik neu entwickelt.

Die Mitglieder der Fachschaft Informatik legen durch ihre überwiegend außerschulisch geprägte Hochschulausbildung und Berufserfahrung im Unterricht besonderen Wert auf Themen wie **Projektmanagement und Teamarbeit** (*Rollen nach Belbin*) sowie praxisnahe Lehrbeispiele. In der Stufe EF werden gezielt Praktika im Bereich der Informatik-Berufe betreut und im Unterricht reflektiert und hier stark mit der Studien- und Berufsorientierung am Gutenberg-Gymnasium kooperiert.

Organisatorisch ist das Fach Informatik in den MINT-Zweig der Schule eingebunden, den Schülerinnen und Schüler als Schwerpunkt in der Oberstufe anwählen können. Die Fachschaft Informatik unterstützt das Bestreben des Gutenberg-Gymnasiums, als **MINT-freundliche Schule** zertifiziert zu werden.

In der **Sekundarstufe II** bietet das Gutenberg-Gymnasium für die eigenen Schülerinnen und Schüler in der Einführungsphase jeweils drei Grundkurse und in der Qualifikationsphase jeweils zwei bis drei Grundkurse in Informatik an. Leistungskurse werden im Fach Informatik am Gutenberg-Gymnasium derzeit nicht angeboten.

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

Der Unterricht der Sekundarstufe II wird mit Hilfe der Programmiersprache **Java** durchgeführt. In der Einführungsphase kommen dabei zusätzlich didaktische Bibliotheken (*basis.jar* und *robot.jar*) zum Einsatz, welche das Erstellen von grafischen Programmen erleichtern und einen schülerorientierten Einstieg in die Programmierung ermöglichen.

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

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

Zurzeit besteht die Fachschaft Informatik des Gutenberg-Gymnasiums aus drei Lehrkräften, denen vier Computerräume mit je 17 Computerarbeitsplätzen zur Verfügung stehen, von denen einer schwerpunktmäßig dem Fach Informatik gewidmet ist. Darüber steht den Schülerinnen und Schülern die Mediothek mit 10 Plätzen zur Verfügung. Alle **Arbeitsplätze** sind an das schulinterne Rechnernetz an den im Frühjahr 2014 installierten logoDIDACT-Schulserver angeschlossen, so dass Schülerinnen und Schüler über einen individuell gestaltbaren Zugang zum zentralen Server der Schule alle Arbeitsplätze der Räume zum Zugriff auf ihre eigenen Daten, zur Recherche im Internet oder zur Bearbeitung schulischer Aufgaben verwenden können. Der Einsatz der Lernplattform Moodle befindet sich in der Erprobung. Die Fachschaft Informatik bringt ihr Know-How im technischen Bereich in besonderem Maße in die Weiterentwicklung der Informations- und Kommunikationsprozesse am Gutenberg-Gymnasium ein.

## 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.

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

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

## 2.1.1 Übersichtsraster Unterrichtsvorhaben

### 1) Qualifikationsphase Q1 (Grundkurs)

Qualifikationsphase – Q1	
<p><b><u>Unterrichtsvorhaben Q1-I</u></b></p> <p><b>Thema:</b> <i>Wiederholung und Vertiefung der objektorientierten Modellierung</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"><li>- Modellieren</li><li>- Darstellen und Interpretieren</li><li>- Implementieren</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>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"><li>- Objekte und Klassen</li><li>- Wirkung der Automatisierung</li></ul> <p><b>Zeitbedarf:</b> 14 Stunden</p>	<p><b><u>Unterrichtsvorhaben Q1-II</u></b></p> <p><b>Thema:</b> <i>Organisation und Verarbeitung von Daten I – Modellierung und Implementierung von Anwendungen mit dynamischen und linearen Datenstrukturen</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>- Argumentieren</li></ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"><li>- Daten und ihre Strukturierung</li><li>- Algorithmen</li></ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"><li>- Objekte und Klassen</li><li>- Syntax und Semantik einer Programmiersprache</li><li>- Analyse, Entwurf und Implementierung von Algorithmen</li><li>- Algorithmen in ausgewählten informatischen Kontexten</li></ul> <p><b>Zeitbedarf:</b> 20 Stunden</p>

## Qualifikationsphase – Q1

### Unterrichtsvorhaben Q1-III

**Thema:**

*Algorithmen zum Suchen und Sortieren auf linearen Datenstrukturen*

**Zentrale Kompetenzen:**

- Argumentieren
- Darstellen und Interpretieren
- Modellieren
- Implementieren

**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:**

*Automaten und formale Sprachen*

**Zentrale Kompetenzen:**

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

**Inhaltsfelder:**

- Formale Sprachen und Automaten
- Informatiksysteme

**Inhaltliche Schwerpunkte:**

- Syntax und Semantik einer Programmiersprache
- Endliche Automaten
- Grammatiken regulärer Sprachen
- Möglichkeiten und Grenzen von Automaten und formalen Sprachen

**Zeitbedarf:** 20 Stunden

**Summe Qualifikationsphase Q1:  
74 Stunden**

## 2) Qualifikationsphase Q2 (Grundkurs)

Qualifikationsphase – Q2	
<p><b><u>Unterrichtsvorhaben Q2-I</u></b></p> <p><b>Thema:</b> <i>Organisation und Verarbeitung von Daten II – Modellierung und Implementierung von Anwendungen mit dynamischen nicht- linearen Datenstrukturen</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"><li>- Argumentieren</li><li>- Darstellen und Interpretieren</li><li>- Modellieren</li><li>- Implementieren</li><li>- Kommunizieren und Kooperieren</li></ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"><li>- Daten und ihre Strukturierung</li><li>- Algorithmen</li><li>- Formale Sprachen und Automaten</li></ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"><li>- Objekte und Klassen</li><li>- Analyse, Entwurf und Implementierung von Algorithmen</li><li>- Algorithmen in ausgewählten informatischen Kontexten</li><li>- Syntax und Semantik einer Programmiersprache</li></ul> <p><b>Zeitbedarf:</b> 20 Stunden</p>	<p><b><u>Unterrichtsvorhaben Q2-II</u></b></p> <p><b>Thema:</b> <i>Aufbau von und Kommunikation in Netzwerken</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"><li>- Argumentieren</li><li>- Darstellen und Interpretieren</li><li>- Kommunizieren und Kooperieren</li></ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"><li>- Informatiksysteme</li><li>- Informatik, Mensch und Gesellschaft</li></ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"><li>- Einzelrechner und Rechnernetzwerke</li><li>- Sicherheit</li><li>- Nutzung von Informatiksystemen, Wirkungen der Automatisierung</li></ul> <p><b>Zeitbedarf:</b> 16 Stunden</p>



## Qualifikationsphase – Q2

### Unterrichtsvorhaben Q2-III

#### **Thema:**

*Nutzung und Modellierung 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:** 20 Stunden

**Summe Qualifikationsphase Q2:  
56 Stunden**

## 2.1.2 Konkretisierte Unterrichtsvorhaben

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

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

Die Schülerinnen und Schüler

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

### 1) Qualifikationsphase Q1 (Grundkurs)

Die hier dargestellten Konkretisierungen beziehen sich auf das Lehrwerk **Informatik 2** und können anhand der Lehrtexte, Projekte und Aufgaben umfassend umgesetzt werden.

#### Unterrichtsvorhaben Q1-I

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

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

#### **Vorhabenbezogene Konkretisierung:**

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

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

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

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

**Zeitbedarf:** 14 Stunden

**Sequenzierung des Unterrichtsvorhabens:** nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Wiederholung der grundlegenden Konzepte der objektorientierten Programmierung</b></p> <p>a) Sichtweise der objektorientierten Informatik auf die Welt</p> <p>b) OOP als informatikspezifische Modellierung der Realität</p> <p>c) Schritte der Softwareentwicklung</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> <li>- analysieren und erläutern objektorientierte Modellierungen (A),</li> <li>- modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M),</li> <li>- ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M),</li> <li>- modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),</li> </ul>	<p>2.1 Die Welt ist voller Objekte  ProjektEinstieg:  Klassenentwurf – step by step</p>
<p><b>2. Erweiterung der objektorientierten Programmierung</b></p> <p>a) Umsetzung einer Anforderung in Entwurfs- und Klassendiagramm</p> <p>b) Objektkommunikation im Sequenzdiagramm</p> <p>c) Klassendokumentation</p> <p>d) Umsetzung von Teilen der Modellierung</p>	<ul style="list-style-type: none"> <li>- nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> <li>- wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I),</li> <li>- stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D),</li> <li>- dokumentieren Klassen (D),</li> <li>- stellen die Kommunikation zwischen Objekten grafisch dar (D),</li> <li>- untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A),</li> <li>- untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A).</li> </ul>	<p>2.2 Gut geplant – Klassenentwurf  2.3 Vererbungshierarchien nutzen</p>
<p><b>3. Mensch und Technik</b></p> <p>a) Verantwortung von Informatikern</p> <p>b) Automatisierung des Alltags durch Informatik</p>	<ul style="list-style-type: none"> <li>- untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A),</li> <li>- untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A).</li> </ul>	<p>Die digitale Welt 001 - Mensch und Technik</p>
<p><b>4. Übung und Vertiefung der OOM / OOP</b></p>	<ul style="list-style-type: none"> <li>- untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A).</li> </ul>	<p>Prüfungsvorbereitung</p>

## Unterrichtsvorhaben Q1-2

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

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

### **Vorhabensbezogene Konkretisierung:**

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

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

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

**Zeitbedarf:** 20 Stunden

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

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Die Datenstruktur Feld</b></p> <p>a) Erarbeitung der Anforderungen an eine Datenstruktur</p> <p>b) Wiederholung der Datenstruktur Array, Eigenschaften der Datenstruktur, Standardoperationen für ein und zweidimensionale Arrays</p> <p>c) Modellierung und Implementierung von Anwendungen</p>	<p>Die Schülerinnen und Schüler...</p> <ul style="list-style-type: none"> <li>- erläutern und begründen methodische Vorgehensweisen, Entwurfs- und Implementationsentscheidungen sowie Aussagen über Informatiksysteme (A)</li> <li>- konstruieren zu kontextbezogenen Problemstellungen informatische Modelle (M)</li> <li>- ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M)</li> <li>- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M),</li> <li>- implementieren auf der Grundlage von Modellen oder Modellausschnitten Computerprogramme (I)</li> <li>- testen und korrigieren Computerprogramme (I)</li> <li>- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>- überführen gegebene textuelle und grafische Darstellungen informatischer Zusammenhänge in die jeweils andere Darstellungsform (D)</li> <li>- stellen informatische Modelle und Abläufe in Texten, Tabellen, Diagrammen und Grafiken dar (D)</li> <li>- stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D)</li> <li>- modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M)</li> <li>- ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M)</li> <li>- dokumentieren Klassen (D)</li> <li>- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klas-</li> </ul>	<p><b>Kapitel:</b></p> <ul style="list-style-type: none"> <li>- Anforderungen an eine Datenstruktur</li> <li>- Speichern mit Struktur</li> </ul> <p><i>neue Wdh. Aufgabe entwickeln, z. B. eine Chart-Top-10, eine Aufgabe mit zweidimensionalem Array (vgl. Anforderungen KLP)</i></p>
<p><b>2. Die Datenstruktur Schlange</b></p> <p>a) Modellierung und Implementierung der Verknüpfung von Objekten</p> <p>b) Generische Typen, Trennung von Verwaltung und Inhalt dyn. DS.</p> <p>c) Erläuterung von Problemstellungen, die nach dem FIFO-Prinzip bearbeitet werden</p> <p>d) Funktionalität der Schlange unter Verwendung der Klasse <i>Queue</i>; Erschließen der Standardoperationen</p> <p>e) Modellierung und Implementierung einer Anwendung auf der Basis einer Anforderungsbeschreibung mit Objekten der Klasse <i>Queue</i></p>	<ul style="list-style-type: none"> <li>- Wer zuerst kommt</li> <li>- Objekte miteinander verketteten</li> <li>- Verwaltung und Inhalt</li> <li>- Funktionen der Queue</li> </ul> <p><b>Aufgaben:</b></p> <ul style="list-style-type: none"> <li>- Warteschlange Büro (Standardoperationen/Basiskompetenz) Kunden warten auf einem Flur, um in ein Büro vorgelassen zu werden. Sie können sich am Ende der Warteschlange anstellen, vorgelassen werden oder müssen alle gehen, wenn die Sprechzeit vorüber ist.</li> <li>- Erweiterte Queue Verkehrskontrolle (Vertiefung) Die Polizei kontrolliert die Fahrzeuge im Hinblick auf ihre Verkehrstauglichkeit. Für die Kontrolle werden die Fahrzeuge aus</li> </ul>	<p><b>Kapitel:</b></p> <ul style="list-style-type: none"> <li>- Wer zuerst kommt</li> <li>- Objekte miteinander verketteten</li> <li>- Verwaltung und Inhalt</li> <li>- Funktionen der Queue</li> </ul> <p><b>Aufgaben:</b></p> <ul style="list-style-type: none"> <li>- Warteschlange Büro (Standardoperationen/Basiskompetenz) Kunden warten auf einem Flur, um in ein Büro vorgelassen zu werden. Sie können sich am Ende der Warteschlange anstellen, vorgelassen werden oder müssen alle gehen, wenn die Sprechzeit vorüber ist.</li> <li>- Erweiterte Queue Verkehrskontrolle (Vertiefung) Die Polizei kontrolliert die Fahrzeuge im Hinblick auf ihre Verkehrstauglichkeit. Für die Kontrolle werden die Fahrzeuge aus</li> </ul>

	senbibliotheken (I)	dem Verkehr gewun- ken. Es werden so lange Fahrzeuge kon- trolliert, bis eine ge- wissen Menge an Verstößen vorliegt oder Autos kontrolliert wurden.
<p><b>3. Die Datenstruktur Stapel</b></p> <p>a) Erläuterung von Problemstellungen, die nach dem LIFO-Prinzip bearbeitet werden</p> <p>b) Funktionalität der Klasse Stapel unter Verwendung der Klasse <i>Stack</i></p> <p>c) Modellierung und Implementierung einer Anwendung auf Basis einer Anforderungsbeschreibung mit Objekten der Klasse <i>Queue</i></p> <p>d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (Objekte der Klassen <i>Queue</i>, <i>Stack</i> und <i>Array (Palindrom)</i>)</p>		<p><b>Kapitel:</b></p> <ul style="list-style-type: none"> <li>- Daten gut abgelegt – Stapel</li> <li>- Funktionen der Datenstruktur Stapel</li> </ul> <p><b>Aufgaben:</b></p> <ul style="list-style-type: none"> <li>- Standardoperationen/Basiskompetenz (Stapel Münzen/ CDs) zur Umsetzung der gegebenen Funktionen der Klasse <i>Stack</i></li> <li>- Biber und Teller Es gibt große und kleine Biber sowie grüne und braune Teller. Es muss überprüft werden, ob die gestapelten Teller zur Schlange der Biber passen, da die großen Biber nur von den braunen Tellern essen und die kleinen von den grünen. Hierbei müssen sowohl <i>Queue</i> als auch <i>Stack</i> verwendet werden.</li> <li>- Palindrom Es wird überprüft, ob ein beliebiges Wort ein Palindrom ist.</li> </ul>
<p><b>4. Die Datenstruktur Liste</b></p> <p>a) Analyse der Möglichkeiten bisheriger Datenstrukturen zwecks Bestimmung notwendiger Funktionalitäten für</p>		<p><b>Kapitel:</b></p> <ul style="list-style-type: none"> <li>- Flexibel für alle Fälle – (die) lineare Liste</li> <li>- Funktionen der Datenstruktur Liste</li> </ul> <p><b>Aufgaben:</b></p>

<p>komplexere Anwendungen (Abgrenzung zu <i>Stack/Queue</i>, zusätzliche Fähigkeiten der Klasse <i>List</i>)</p> <p>b) Erarbeitung der Funktionalität der Liste unter Verwendung der Klasse <i>List</i></p> <p>c) Modellierung und Implementierung einer Anwendung mit Objekten der Klasse <i>List</i></p> <p>d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (Stack, Queue, List)</p>		<ul style="list-style-type: none"> <li>- LEDs</li> <li>- Textzeilen verarbeiten</li> </ul>
<p><b>5. Übungen und Vertiefungen zur Verwendung linearer und dynamischer Datenstrukturen anhand weiterer Problemstellungen</b></p>		<p>Prüfungsvorbereitung</p>
<p><b>Alternative:</b></p>		<p><i>Projekteinstieg Heldenspiel</i></p> <p><i>Mit dem Heldenspiel können alle im Kapitel behandelten Datenstrukturen erarbeitet werden. Das Spiel kann bis zu einem beliebigen Grad realisiert werden, sodass es sowohl als Einstieg als auch als ein umfassendes Projekt für lineare Datenstrukturen genutzt werden kann.</i></p>

## **Unterrichtsvorhaben Q1-III**

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

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

### **Vorhabenbezogene Konkretisierung:**

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

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

In der zweiten Unterrichtssequenz geht es um die Frage, wie Daten in linearen Strukturen (lineare Liste und Array) (wieder-)gefunden werden können. Die lineare Suche als iteratives und die binäre Suche als rekursives Verfahren werden veranschaulicht und implementiert. Die Bewertung der Algorithmen erfolgt, indem jeweils die Anzahl der Vergleichsoperationen und der Speicherbedarf ermittelt wird.

Möchte man Daten effizient in einer linearen Struktur wiederfinden, so rückt zwangsläufig die Frage nach einer Sortierstrategie in den Fokus. Es wird mindestens ein iteratives und ein rekursives Sortierverfahren erarbeitet und implementiert sowie ihre Effizienz bewertet.

**Zeitbedarf:** 20 Stunden

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



Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Eigenschaften von Algorithmen</b></p> <p>a) Qualitätseigenschaften von Algorithmen</p> <p>b) Strukturierung von Algorithmen mit Hilfe der Strategien „Modularisierung“ und „Teile und Herrsche“</p> <p>c) Analyse und Entwicklung von rekursiven Algorithmen</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>- analysieren und erläutern Algorithmen und Programme (A),</li> <li>- modifizieren Algorithmen und Programme (I),</li> <li>- stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D),</li> <li>- entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M),</li> </ul>	<p>4.1 Ohne Algorithmen läuft nichts</p> <p>4.2 Teile die Arbeit – rekursive Algorithmen</p>
<p><b>2. Suchen in Listen und Arrays</b></p> <p>a) Lineare Suche in Listen und Arrays</p> <p>b) Binäre Suche in einem Array</p> <p>c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf</p>	<ul style="list-style-type: none"> <li>- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),</li> <li>- testen Programme systematisch anhand von Beispielen (I).</li> <li>- implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I),</li> <li>- beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A),</li> </ul>	<p>4.3 Suchen – iterativ und rekursiv</p>
<p><b>3. Sortieren auf Listen und Arrays</b></p> <p>a) Entwicklung und Implementierung eines iterativen Sortierverfahrens für eine Liste</p> <p>b) Entwicklung und Implementierung eines rekursiven Sortierverfahrens für ein Array</p> <p>c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf</p>	<ul style="list-style-type: none"> <li>- beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li> <li>- nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> <li>- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>- testen Programme systematisch anhand von Beispielen (I),</li> </ul>	<p>4.4 Sortieren – iterativ und rekursiv</p>

## Unterrichtsvorhaben Q-I IV

**Thema:** *Automaten und formale Sprachen*

### **Leitfragen:**

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

### **Vorhabensbezogene Konkretisierung:**

Ausgehend von der Beschreibung und Untersuchung realer Automaten wird das formale Modell eines endlichen Automaten entwickelt. Neben dem Mealy-Automaten geht es vor allem um den erkennenden endlichen Automaten. Auf die Erarbeitung der Beschreibung folgt die Modellierung

eigener Automaten und die Untersuchung bestehender, um die Eigenschaften und Grenzen eines endlichen Automaten zu erkennen. Hierbei wird dessen Verhalten auf bestimmte Eingaben analysiert.

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

**Zeitbedarf:** 20 Std.

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

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Endliche Automaten</b></p> <p>a) Erarbeitung der formalen Beschreibung eines Mealy-Automaten und der Darstellungsformen</p> <p>b) Erarbeitung der formalen Beschreibung eines deterministischen endlichen Automaten (DEA) sowie dessen Darstellungsformen; Erschließung der Fachbegriffe Alphabet, Wort, (akzeptierte) Sprache, Determinismus</p> <p>c) Analyse der Eigenschaften von DEA durch die Modellierung eines Automaten zu einer gegebenen Problemstellung, der Modifikation eines Automaten sowie die Überführung der gegebenen Darstellungsform in eine andere</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>- stellen informatische Modelle und Abläufe in Texten, Tabellen, Diagrammen und Grafiken dar (D)</li> <li>- analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens bei bestimmten Eingaben (A)</li> <li>- ermitteln die Sprache, die ein endlicher Automat akzeptiert (D)</li> <li>- entwickeln und modifizieren zu einer Problemstellung endlicher Automaten (M)</li> <li>- stellen endliche Automaten in Tabellen und Graphen dar und überführen sie in die jeweils andere Darstellungsform (D)</li> <li>- entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M)</li> <li>- analysieren und erläutern Grammatiken regulärer Sprachen (A)</li> <li>- modifizieren Grammatiken regulärer Sprachen (M)</li> <li>- entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M)</li> </ul>	<p>Vom realen Automaten zum Modell</p> <p>Projekteinstieg: Schatzsuche Der Mealy-Automat Der erkennende endliche Automat Wort und Sprache</p>
<p><b>2. Grammatiken regulärer Sprachen</b></p> <p>a) Erarbeitung der formalen Beschreibung einer regulären Grammatik (Sprache, Terminal und Nicht-Terminal, Produktionen und Produktionsvorschriften)</p> <p>b) Analyse der Eigenschaften einer regulären Grammatik durch deren Entwicklung und Modellierung zu einer gegebenen Problemstellung.</p>	<ul style="list-style-type: none"> <li>- entwickeln zur akzeptierten Sprache eines Automaten eine zugehörige Grammatik (M)</li> <li>- beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D)</li> <li>- zeigen die Grenzen endlicher Automaten und regulärer Grammatiken auf</li> </ul>	<p>Grammatiken regulärer Sprachen</p>
<p><b>3. Übungen und Vertiefungen</b></p> <p>Verwendung endlicher Automaten und</p>		<p>Prüfungsvorbereitung</p>

## 2) Qualifikationsphase Q2 (Grundkurs)

### Unterrichtsvorhaben Q2-I

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

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

#### **Vorhabenbezogene Konkretisierung:**

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

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

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

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

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

**Zeitbedarf:** 20 Stunden

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

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Aufbau von Baumstrukturen und Grundbegriffe</b></p> <p>a) Erarbeitung der Begriffe Wurzel, Knoten, Blatt, Kante, Grad eines Knotens und eines Baumes, Pfad, Tiefe, Ebene, Teilbaum</p> <p>b) Aufbau und Darstellung von Baumstrukturen in verschiedenen Anwendungskontexten</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>- stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D),</li> <li>- erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A),</li> <li>- analysieren und erläutern Algorithmen und Programme (A),</li> <li>- stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).</li> </ul>	<p>6.1 Spielen mit Struktur – Baumstrukturen</p> <p>Projekteinstieg 1: Spielbäume</p>
<p><b>2. Binäre Bäume</b></p> <p>a) rekursiver Aufbau eines binären Baums</p> <p>b) Traversierungen (pre-, in-, postorder)</p> <p>c) Modellierung eines Binärbaums in einem Anwendungskontext mit Hilfe der Klasse BinaryTree (als Entwurfs- und Implementationsdiagramm)</p> <p>d) Implementation einer Anwendung der Datenstruktur binärer Baum (ggf. in Teilen)</p>	<ul style="list-style-type: none"> <li>- beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li> <li>- ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li> <li>- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M),</li> <li>- modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),</li> <li>- verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M),</li> </ul>	<p>6.2 Zwei Nachfolger sind genug! - Binäre Bäume</p> <p>Implementation des Projekts Ahnenbaum</p>
<p><b>3. Binäre Suchbäume</b></p> <p>a) Prinzip des binären Suchbaums, Ordnungsrelation</p> <p>b) Operationen auf dem binären Suchbaum (Suchen, Einfügen, Löschen, sortierte Ausgabe)</p> <p>c) Modellierung eines binären Suchbaums in einem Anwendungskontext mit Hilfe der Klasse BinarySearchTree (als Entwurfs- und Implementationsdiagramm)</p>	<ul style="list-style-type: none"> <li>- entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M),</li> <li>- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),</li> <li>- modifizieren Algorithmen und Programme (I),</li> <li>- nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> </ul>	<p>6.3 Wer Ordnung hält, spart Zeit beim Suchen – Binäre Suchbäume</p> <p>Projekteinstieg 2: Binäre Suchbäume</p> <p>Implementation des Projekts Benutzerverwaltung</p>

<p>m) und dem Interface Item</p> <p>d) Implementation einer Anwendung der Datenstruktur binärer Suchbaum (ggf. in Teilen)</p>	<ul style="list-style-type: none"> <li>- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>- testen Programme systematisch anhand von Beispielen (I),</li> </ul>	
<p><b>4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen</b></p>		<p>Prüfungsvorbereitung</p>

## Unterrichtsvorhaben Q2-II

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

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

### **Vorhabenbezogene Konkretisierung:**

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

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

**Zeitbedarf:** 12 Stunden

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

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Technische Kommunikation als Fortführung natürlicher Kommunikation</b></p> <p>a) Kommunikation im Shannon-Weaver-Modell</p> <p>b) Kriterien von technischen Kommunikationsarten</p> <p>c) Die Geschichte der technischen Kommunikation</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>- beschreiben und erläutern Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A),</li> <li>- nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D),</li> <li>- analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A).</li> </ul>	<p>7.1 Menschen kommunizieren – ohne und mit Technik</p> <p>Projekteinstieg: Kommunikation im Wilden Westen</p>
<p><b>2. Aufbau von Netzwerken und Kommunikationsregeln</b></p> <p>a) Das Netzwerk als Organisationsprinzip der Kommunikation und Möglichkeiten der Ausformung</p> <p>b) Geregeltete technische Kommunikation durch Protokolle in Schichtenmodellen</p>		<p>7.2 Ohne Protokoll läuft nichts - Netzwerke</p>
<p><b>3. Aufgabenteilung in Netzwerken durch Server und Client</b></p> <p>a) Aufbau und Aufgaben der Client-Server-Struktur</p> <p>(b) Protokolle zwischen Client und Server</p>		<p>7.3 Einer für alle – Client-Server-Struktur</p>
<p><b>4. Kryptologie</b></p> <p>a) Veranschaulichen und Anwenden von symmetrischen und asymmetrischen kryptographischen Verfahren (Caesar, Vigenère, RSA)</p> <p>b) Bewertung der Verfahren hinsichtlich ihrer Sicherheit und ihrem Aufwand</p>		<p>Die digitale Welt 100 – Kryptologie</p>

### **Unterrichtsvorhaben Q2-3**

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

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

**Vorhabenbezogene Konkretisierung:**

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

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

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

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

**Zeitbedarf:** 20 Stunden

**Sequenzierung des Unterrichtsvorhabens:** nächste Seite



Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Nutzung von relationalen Datenbanken</b></p> <p>a) Aufbau von Datenbanksystemen und Grundbegriffe</p> <ul style="list-style-type: none"> <li>- Aufgaben und Eigenschaften eines Datenbanksystems</li> <li>- Erarbeitung der Begriffe Tabelle, Attribut, Attributwert, Datensatz, Datentyp, Primärschlüssel, Datenbankschema</li> <li>- Problematisierung von Redundanzen, Anomalien und Inkonsistenzen</li> </ul> <p>b) SQL-Abfragen</p> <ul style="list-style-type: none"> <li>- Erarbeitung der grundlegenden Sprachelemente von SQL (SELECT(DISTINCT), FROM, WHERE, JOIN)</li> <li>- Analyse und Erarbeitung von SQL-Abfragen (AND, OR, NOT, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, &lt;&gt;, &gt;, &lt;, &gt;=, &lt;=, LIKE, BETWEEN, IN, IS NULL, geschachtelte Select-Ausdrücke)</li> </ul> <p>c) Vertiefung an einem weiteren Datenbankbeispiel</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>- erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A),</li> <li>- analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A),</li> <li>- analysieren und erläutern eine Datenbankmodellierung (A),</li> <li>- erläutern die Eigenschaften normalisierter Datenbankschemata (A),</li> <li>- bestimmen Primär- und Sekundärschlüssel (M),</li> <li>- ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M),</li> <li>- modifizieren eine Datenbankmodellierung (M),</li> <li>- modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M),</li> <li>- überführen Datenbankschemata in vorgegebene Normalformen (M),</li> <li>- verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I),</li> <li>- ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D),</li> <li>- stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D),</li> <li>- überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D).</li> </ul>	<p><b>Kapitel 8 Datenbanken</b></p> <p>8.1 Wissen speichern und verwalten – Datenbanksysteme</p> <p>8.2 Daten anordnen mit Tabellen</p> <p>Beispiel: Buchhandlung</p> <p>Redundanzen, Anomalien und Inkonsistenzen</p> <p>Projekteinstieg: Tabellen</p> <p>8.3 Daten filtern mit SQL</p> <p>8.4 Komplexe Filter</p>

<p><b>2. Modellierung von relationalen Datenbanken</b></p> <p>a) Datenbankentwurf durch ER-Diagramme</p> <ul style="list-style-type: none"> <li>- Ermittlung von Entitäten, zugehörigen Attributen, Beziehungen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms</li> <li>- Erläuterung und Erweiterung einer Datenbankmodellierung</li> </ul> <p>b) Entwicklung eines relationalen Modells aus einem Datenbankentwurf</p> <ul style="list-style-type: none"> <li>- Überführung eines Entity-Relationship-Diagramms in ein relationales Datenbankschema inklusive der Bestimmung von Primär- und Fremdschlüsseln</li> </ul> <p>c) Normalformen</p> <ul style="list-style-type: none"> <li>- Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten)</li> </ul>		<p>8.5 Datenbankentwurf</p> <p><i>Beispiel: Online-Buchhandel</i></p> <p><i>Datenanalyse und Entwurf</i></p> <p>8.6 Umsetzung des ER-Modells</p> <p><i>Entitätsmengen</i></p> <p><i>m:n-Beziehungen</i></p> <p><i>1:n-Beziehungen</i></p> <p><i>1:1-Beziehungen</i></p> <p><i>Wiederaufgriff des Projekteinstiegs</i></p> <p>8.7 Datenbanken verbessern durch Normalformen</p>
<p><b>3. Übung und Vertiefung der Nutzung und Modellierung von relationalen Datenbanken</b></p>		<p>Prüfungsvorbereitung</p>

## 2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

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

### Überfachliche Grundsätze:

- 1) Geeignete Problemstellungen zeichnen die Ziele des Unterrichts vor und bestimmen die Struktur der Lernprozesse.
- 2) Inhalt und Anforderungsniveau des Unterrichts sind auf das Alter der Lerngruppen ebenso abgestimmt sind wie auf den erreichten Stand der Kompetenzen der individuellen Lerngruppe.
- 3) Die Unterrichtsgestaltung ist auf die Ziele und Inhalte abgestimmt.
- 4) Medien und Arbeitsmittel sind schülernah gewählt.
- 5) Die Schülerinnen und Schüler erreichen einen Lernzuwachs.
- 6) Der Unterricht fördert eine aktive Teilnahme der Schülerinnen und Schüler und führt sie schrittweise zur Eigenverantwortung als autonom Lernende.
- 7) Der Unterricht fördert die Zusammenarbeit zwischen den Schülerinnen und Schülern und bietet ihnen Möglichkeiten zu eigenen Lösungen.
- 8) Der Unterricht versucht individuelle Lernwege zu berücksichtigen. Wann immer es möglich ist, bietet der Unterricht Möglichkeiten der Binnendifferenzierung.
- 9) Die Schülerinnen und Schüler erhalten Gelegenheit zu selbstständiger Arbeit und werden dabei unterstützt.
- 10) Der Unterricht fördert strukturierte und funktionale Partner- bzw. Gruppenarbeit.
- 11) Der Unterricht fördert strukturierte und funktionale Arbeit im Plenum.
- 12) Die Lernumgebung ist vorbereitet; der Ordnungsrahmen wird eingehalten.
- 13) Die Lehr- und Lernzeit wird intensiv für Unterrichtszwecke genutzt.
- 14) Es herrscht ein positives pädagogisches Klima im Unterricht.
- 15) Die Einhaltung und Weiterentwicklung der genannten Perspektiven erfolgt in innerkollegialen Hospitationsphasen, die jeweils thematisch fokussiert werden in jedem Halbjahr.

### Fachliche Grundsätze:

- 1) Der Unterricht unterliegt der Wissenschaftsorientierung und ist dementsprechend eng verzahnt mit seiner Bezugswissenschaft.
- 2) Der Unterricht ist problemorientiert und soll von realen Problemen ausgehen und sich auf solche rückbeziehen.
- 3) Der Unterricht folgt dem Prinzip der Exemplarizität und soll ermöglichen, informatische Strukturen und Gesetzmäßigkeiten in den ausgewählten Problemen und Projekten zu erkennen.
- 4) Der Unterricht ist anschaulich sowie gegenwarts- und zukunftsorientiert und gewinnt dadurch für die Schülerinnen und Schüler an Bedeutsamkeit.
- 5) Der Unterricht ist handlungsorientiert, d.h. projekt- und produktorientiert angelegt.
- 6) Im Unterricht werden sowohl für die Schule didaktisch reduzierte als auch reale Informatiksysteme aus der Wissenschafts-, Berufs- und Lebenswelt eingesetzt.
- 7) 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 Informatik des Gutenberg-Gymnasiums im Einklang mit dem entsprechenden schulbezogenen Konzept die nachfolgenden Grundsätze zur Leistungsbewertung und Leistungsrückmeldung beschlossen. Die nachfolgenden Absprachen stellen die Minimalanforderungen an das lerngruppenübergreifende gemeinsame Handeln der Fachgruppenmitglieder dar. Bezogen auf die einzelne Lerngruppe kommen ergänzend weitere der in den Folgeabschnitten genannten Instrumente der Leistungsüberprüfung zum Einsatz.

### 2.3.1 Beurteilungsbereich Klausuren und Facharbeiten

Die Beurteilung der Schülerinnen und Schüler mit schriftlicher Belegung erfolgt zusätzlich zur sonstigen Mitarbeit auf Basis der in schriftlichen Klausuren erbrachten Leistungen, wobei die erste Klausur in der Q1.2 durch eine Facharbeit ersetzt werden kann.

#### 1) 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.

Die Klausuren werden wie in anderen Fächern auch auf Papier ohne Einsatz von Computern geschrieben.

##### **Instrumente:**

- Einführungsphase: 1 Klausur je Halbjahr  
Dauer der Klausur: 2 Unterrichtsstunden
- Grundkurse Q 1: 2 Klausuren je Halbjahr  
Dauer der Klausuren: 3 Unterrichtsstunden
- Grundkurse Q 2.1: 2 Klausuren  
Dauer der Klausuren: 3 Unterrichtsstunden
- Grundkurse Q 2.2: 1 Klausur unter Abiturbedingungen
- Anstelle einer Klausur kann gemäß dem Beschluss der Lehrerkonferenz in Q 1.2 eine Facharbeit geschrieben werden.

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

##### **Kriterien**

Die Bewertung der schriftlichen Leistungen in Klausuren erfolgt über ein Raster mit Hilfspunkten, die im Erwartungshorizont den einzelnen Kriterien zugeordnet sind. Spätestens ab der Qualifikationsphase orientiert sich die Zuordnung der Hilfspunktsomme zu den Notestufen 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) Facharbeiten

Die erste Klausur im 2. Halbjahr kann nach Absprache mit den Tutoren und dem Fachlehrer sowie abhängig von den Kapazitäten (Facharbeiten pro Kurslehrer) durch eine Facharbeit ersetzt werden.

Im Fach Informatik besteht neben empirischen Arbeiten und Literaturarbeiten insbesondere die *Möglichkeit, eine Arbeit mit Bezug zur Softwareentwicklung* durchzuführen. Damit stehen drei Arten von Facharbeiten zur Auswahl:

- a. Softwareentwicklung-Arbeit (Entwicklung und Dokumentation einer Java-Software)
- b. Empirisch-experimentelle Arbeit (Modellierung, Erhebung und Auswertung)
- c. Literaturarbeit (Arbeit mit Fachliteratur)

### Verbindliche Absprachen:

Es gilt jeweils die für den jeweiligen Jahrgang auf [www.gugy.de](http://www.gugy.de) veröffentlichte „Handreichung zur Facharbeit in der Jahrgangsstufe Q1“ des Gutenberg-Gymnasiums.

Im ersten Halbjahr der Q1 erfolgt eine Information durch den Fachlehrer über die Möglichkeiten und Anforderungen einer Facharbeit in Informatik und die Vorstellung von Beispielthemen.

### Instrumente:

- Grundkurse Q 1: 1 Facharbeit im 2. Halbjahr als Ersatz für eine Kursarbeit

### Kriterien

Die Bewertung erfolgt auf Basis der Kriterien zur Facharbeit aus dem allgemeinen Leistungskonzept des Gutenberg-Gymnasiums. Bei Softwareentwicklungs-Arbeiten werden die Schritte und Kriterien für gute Softwareentwicklung (vgl. Lehrbuch Q1, S. 15) zugrunde gelegt.

Die Rückmeldung erfolgt in Form eines Bewertungsschemas inkl. Zusammenfassender Beurteilung.

## 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

- Alle Schülerinnen und Schüler führen in der Einführungsphase in Kleingruppen ein Kurzprojekt durch und fertigen dazu eine Arbeitsmappe mit Arbeitstagebuch an. Dies wird in die Note für die Sonstige Mitarbeit einbezogen.
- In 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

### Praktische Leistungen am Computer

- Implementierung, Test und Anwendung von Informatiksystemen

### Sonstige schriftliche Leistungen

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

## **Kriterien**

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

Die Bewertungskriterien stützen sich auf

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

Besonderes Augenmerk ist dabei auf

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

Bei Gruppenarbeiten auch auf

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

Bei Projektarbeit darüber hinaus auf

- die Dokumentation des Arbeitsprozesses,
- den Grad der Selbstständigkeit,

- die Reflexion des eigenen Handelns und
- die Aufnahme von Beratung durch die Lehrkraft.

## **Grundsätze der Leistungsrückmeldung und Beratung**

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

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

Die Leistungsrückmeldung kann

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

erfolgen.

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

## 2.4 Lehr- und Lernmittel

Zum Schuljahr 2013/2014 wurde für die Stufe EF das Lehrwerk „Informatik 1 - Softwareentwicklung mit Greenfoot und BlueJ“ des Schöningh Verlags eingeführt. Es bildet die Grundlage der gemeinsamen Arbeit. In der Qualifikationsphase wird ab dem Schuljahr 2015/2016 der Folgeband „Informatik 2“ eingesetzt.

Ergänzend werden teils durch die Fachschaft selbstverfasste Skripte und Übungen zur Programmiersprache Java verwendet. Die in Raum 217 befindliche Informatik-Bibliothek kann sowohl für die Unterrichtsvorbereitung als auch für Recherchen von Schülerinnen und Schülern genutzt werden. Darüber hinaus werden gezielt Online-Ressourcen wie die Angebote von Daniel Garmann oder [www.swisseduc.ch](http://www.swisseduc.ch) eingesetzt.

Auf allen Schul-PCs sind die benötigten Softwareentwicklungsumgebungen Greenfoot (inkl. Robot-Szenario) und BlueJ (inkl. Basis-Bibliothek) installiert. Sie stehen interessierten Schülerinnen und Schülern auch kostenlos zum Download zur Verfügung.

Die moderne logoDIDACT-Umgebung bietet zusätzlich weitere Informatik-spezifische Software wie UML-Modellierungssoftware und umfangreiche Basis-Software wie das MS Office 2010 Professionell Paket, Mindmap-Software, E-Mail-Clients, Browser etc. und Kollaborationsmöglichkeiten für eine professionelle Arbeitsumgebung.

Der schwerpunktmäßig für das Fach Informatik genutzte Raum 216 bietet neben 17 PC-Arbeitsplätzen und zusätzlichen Gruppenarbeitstischen einen modernen Beamer und ein Multimedia-Soundsystem, ein White-Board, einen Flipchart, einen Time-Timer sowie zahlreiche Themenplakate zu prägenden „Gesichtern der Informatik“.



### **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, soll eine mögliche Zusammenarbeit mit den Fächern Sozialwissenschaften und Philosophie in einer gemeinsamen Fachkonferenz ausgelotet werden.

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

Möglichst schon im zweiten Halbjahr der Einführungsphase, spätestens jedoch im ersten 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, sowie Übungen zum wissenschaftlichen Arbeiten. Es wird angestrebt, dass nur Facharbeiten vergeben werden, die mit der eigenständigen Entwicklung eines Softwareproduktes verbunden sind.

#### **Exkursionen**

In der Einführungsphase soll eine Exkursion zum Schülerlabor der RWTH Aachen durchgeführt werden. Die außerunterrichtliche Veranstaltung wird im Unterricht vor- und nachbereitet.

## **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.

Das schulinterne Curriculum (siehe 2.1) ist zunächst bis 2017 für den ersten Durchgang durch die gymnasiale Oberstufe nach Erlass des Kernlehrplanes verbindlich. Nach jedem Schuljahr werden in einer Sitzung der Fachkonferenz Erfahrungen ausgetauscht und ggf. Änderungen für den nächsten Durchgang der Einführungsphase beschlossen, um erkannten ungünstigen Entscheidungen schnellstmöglich entgegenwirken zu können.

Nach Abschluss des Abiturs 2017 wird die Fachkonferenz Informatik auf der Grundlage ihrer Unterrichtserfahrungen eine Gesamtsicht des schulinternen Curriculums vornehmen und ggf. eine Beschlussvorlage für die erste Fachkonferenz des folgenden Schuljahres erstellen.