



EVANGELISCHES GYMNASIUM LIPPSTADT

BECKUMER STRASSE 61  
59555 LIPPSTADT

FACHSCHAFT INFORMATIK

---

## **Curriculum Informatik**

---

17. Februar 2019 (Version 2)  
verbindlich ab Schuljahr 2016/17

# 1 Die Fachschaft Informatik des Evangelischen Gymnasiums Lippstadt

Beim Evangelischen Gymnasium Lippstadt (EG) handelt es sich um eine dreizügige Schule mit zurzeit ca. 700 Schülerinnen und Schülern, 43 Planstellen und 60 Lehrerinnen und Lehrern. Das Einzugsgebiet der Schule umfasst Lippstadt und angrenzende kleinere Städte und Ortschaften. Im Bereich der Sekundarstufe II kooperiert das Evangelische Gymnasium mit seinen Nachbarschulen und bietet mit ihnen zahlreiche gemeinsame Kurse an, unter anderem in Informatik.

Das Fach Informatik wird am EG ab der Jahrgangsstufe 8 im Wahlpflichtbereich II (WP II) zweistündig unterrichtet und von etwa der Hälfte der Schülerinnen und Schüler besucht. Aufgrund der hohen Nachfrage und unterschiedlicher Lernvoraussetzungen werden zwei niveaudifferente Kurse angeboten. Der eine Kurs legt den Schwerpunkt auf die Gestaltung von Produkten mit informatischen Werkzeugen und vermittelt Grundlagen des Faches Informatik wie z.B. Algorithmik, Daten, Objekte und Kryptologie. Ein zweiter Kurs wurde speziell für Schülerinnen und Schüler mit erweiterten Vorkenntnissen eingerichtet. Im Rahmen dieses Kurses wird auf die Gestaltung mit informatischen Werkzeugen (Office-Programme) verzichtet und die Grundlagen der Informatik vertieft sowie weitere exemplarische vertiefende Themen angesprochen. Durch diese Trennung sollen die Vorkenntnisse und Interessen der Schülerinnen und Schüler berücksichtigt werden und so eine noch bessere individuelle Förderung erlauben.

Organisatorisch ist das Fach Informatik in der Sekundarstufe I in den MINT-Zweig der Schule eingebunden, den Schülerinnen und Schüler als Alternative zu einem bilingualen Zweig anwählen können.

In der Sekundarstufe II bietet das EG für die eigenen Schülerinnen und Schüler in allen Jahrgangsstufen jeweils einen Grundkurs in Informatik an. Ein Leistungskurs kann durch Kooperation mit den anderen Lippstädter Schulen angeboten werden.

Um insbesondere Schülerinnen und Schülern gerecht zu werden, die in der Sekundarstufe I keinen Informatikunterricht besucht haben, wird in Kursen der Einführungsphase besonderer Wert darauf gelegt, dass keine Vorkenntnisse aus der Sekundarstufe I zum erfolgreichen Durchlaufen des Kurses erforderlich sind. Der Unterricht der Sekundarstufe II wird mit Hilfe der Programmiersprache Java durchgeführt. In der Einführungsphase kommt dabei zusätzlich eine didaktische Bibliothek zum Einsatz, welche das Erstellen von grafischen Oberflächen erleichtert.

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

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

Zurzeit besteht die Fachschaft Informatik des EG aus zwei Lehrkräften, denen drei Computerräume mit insgesamt 72 Computerarbeitsplätzen 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 nach dem reduzierten Doppelstundenmodell (5. und 6. Stunde als Einzelstunde).

## 2 Entscheidungen zum Unterricht

### 2.1 Unterrichtsvorhaben

Die Darstellung der Unterrichtsvorhaben im schulinternen Lehrplan besitzt den Anspruch, sämtliche im Kernlehrplan angeführten Kompetenzen abzudecken. Dies entspricht der Verpflichtung jeder Lehrkraft, Schülerinnen und Schülern Lerngelegenheiten zu ermöglichen, so dass alle Kompetenzerwartungen des Kernlehrplans von ihnen erfüllt werden können. Die entsprechende Umsetzung erfolgt auf zwei Ebenen: der Übersichts- und der Konkretisierungsebene.

Im „Übersichtsraster Unterrichtsvorhaben“ wird die für alle Lehrerinnen und Lehrer gemäß Fachkonferenzbeschluss verbindliche Verteilung der Unterrichtsvorhaben dargestellt. Das Übersichtsraster dient dazu, den Kolleginnen und Kollegen einen schnellen Überblick über die Zuordnung der Unterrichtsvorhaben zu den einzelnen Jahrgangsstufen sowie den im Kernlehrplan genannten Kompetenzen, Inhaltsfeldern und inhaltlichen Schwerpunkten zu verschaffen. Der ausgewiesene Zeitbedarf versteht sich als grobe Orientierungsgröße, die nach Bedarf über- oder unterschritten werden kann. 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. 80 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“ 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.

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

### 2.2 Übersichtsraster Unterrichtsvorhaben

#### 2.2.1 Wahlpflichtkurs bzw. Differenzierungskurs Informatik – Jahrgangsstufe 8 und 9

Unterrichtsvorhaben Diff-8-1	Unterrichtsvorhaben Diff-8-2
<p>Thema: Grundbegriffe der Informatik und Datenverarbeitung</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> <li>• Argumentieren</li> <li>• Darstellen und Interpretieren</li> <li>• Kommunizieren und Kooperieren</li> </ul>	<p>Thema: Einführung Algorithmik anhand der Entwicklung einfacher Spiele mit Scratch</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> <li>• Implementieren</li> <li>• Kommunizieren und Kooperieren</li> <li>• Argumentieren</li> </ul>

<p>Inhaltsfelder:</p> <ul style="list-style-type: none"> <li>• Daten und ihre Strukturierung</li> <li>• Informatiksysteme</li> <li>• Informatik, Mensch und Gesellschaft</li> </ul> <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> <li>• Einzelrechner</li> <li>• Digitalisierung</li> <li>• Einsatz von Informatiksystemen</li> <li>• Einzelrechner und Rechnernetzwerke</li> </ul> <p>Zeitbedarf: 20 Stunden</p>	<p>Inhaltsfelder:</p> <ul style="list-style-type: none"> <li>• Algorithmen</li> <li>• Daten und ihre Strukturierung</li> </ul> <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> <li>• Analyse, Entwurf und Implementierung einfacher Algorithmen</li> <li>• Syntax und Semantik einer Programmiersprache</li> </ul> <p>Zeitbedarf: 20 Stunden</p>
---	--

<p>Unterrichtsvorhaben Diff-8-3</p>	<p>Unterrichtsvorhaben Diff-8-4</p>
<p>Thema: Sichere Kommunikation in Rechnernetzen und insbesondere dem Internet</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> <li>• Kommunizieren und Kooperieren</li> <li>• Argumentieren</li> <li>• Darstellen und Interpretieren</li> </ul> <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> <li>• Informatik, Mensch und Gesellschaft</li> <li>• Informatiksysteme</li> <li>• Daten und ihre Strukturierung</li> </ul> <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> <li>• Einzelrechner und Rechnernetzwerke</li> <li>• Internet</li> <li>• Sicherheit</li> </ul> <p>Zeitbedarf: 20 Stunden</p>	<p>Thema: Entwicklung und Gestaltung von Webseiten</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> <li>• Modellieren</li> <li>• Implementieren</li> <li>• Kommunizieren und Kooperieren</li> </ul> <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> <li>• Informatik, Mensch und Gesellschaft</li> <li>• Formale Sprachen und Automaten</li> <li>• Daten und ihre Strukturierung</li> </ul> <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> <li>• Syntax und Semantik einer Programmiersprache</li> <li>• Nutzung von Informatiksystemen</li> </ul> <p>Zeitbedarf: 20 Stunden</p>

<p>Unterrichtsvorhaben Diff-9-1</p>	<p>Unterrichtsvorhaben Diff-9-2</p>
<p>Thema: Modellierung, Entwicklung von Datenbanken sowie deren Benutzung</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> <li>• Modellieren</li> <li>• Implementieren</li> <li>• Darstellen und Interpretieren</li> </ul>	<p>Thema: Datenbanken-Projekt</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> <li>• Modellieren</li> <li>• Implementieren</li> <li>• Darstellen und Interpretieren</li> </ul>

<p>Inhaltsfelder:</p> <ul style="list-style-type: none"> <li>• Formale Sprachen und Automaten</li> <li>• Daten und ihre Strukturierung</li> <li>• Informatik, Mensch und Gesellschaft</li> </ul> <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> <li>• Datenbanken</li> <li>• Syntax und Semantik einer Programmiersprache</li> <li>• Nutzung von Informatiksystemen</li> </ul> <p>Zeitbedarf: 25 Stunden</p>	<p>Inhaltsfelder:</p> <ul style="list-style-type: none"> <li>• Formale Sprachen und Automaten</li> <li>• Daten und ihre Strukturierung</li> <li>• Informatik, Mensch und Gesellschaft</li> </ul> <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> <li>• Datenbanken</li> <li>• Syntax und Semantik einer Programmiersprache</li> <li>• Nutzung von Informatiksystemen</li> </ul> <p>Zeitbedarf: 15 Stunden</p>
---	---

<p>Unterrichtsvorhaben Diff-9-3</p>	<p>Unterrichtsvorhaben Diff-9-4</p>
<p>Thema: Grundlagen Objekte und Algorithmik</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> <li>• Darstellen und Interpretieren</li> <li>• Modellieren</li> <li>• Argumentieren</li> </ul> <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> <li>• Algorithmen</li> <li>• Daten und ihre Strukturierung</li> </ul> <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> <li>• Objekte</li> <li>• Algorithmen in ausgewählten informatischen Kontexten</li> </ul> <p>Zeitbedarf: 20 Stunden</p>	<p>Thema: Einführung in objektorientiertes Programmieren</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> <li>• Implementieren</li> <li>• Modellieren</li> <li>• Kommunizieren und Kooperieren</li> </ul> <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> <li>• Algorithmen</li> <li>• Daten und ihre Strukturierung</li> <li>• Formale Sprachen und Automaten</li> </ul> <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> <li>• Syntax und Semantik einer Programmiersprache</li> <li>• Analyse, Entwurf und Implementierung einfacher Algorithmen</li> </ul> <p>Zeitbedarf: 20 Stunden</p>

**2.2.2 Einführungsphase – EF**

<p>Unterrichtsvorhaben EF-1</p>	<p>Unterrichtsvorhaben EF-2</p>
<p>Thema: Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> <li>• Argumentieren</li> <li>• Darstellen und Interpretieren</li> <li>• Kommunizieren und Kooperieren</li> </ul>	<p>Thema: Grundlagen der objektorientierten Analyse und Modellierung</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> <li>• Modellieren</li> <li>• Darstellen und Interpretieren</li> <li>• Kommunizieren und Kooperieren</li> </ul>

<p>Inhaltsfelder:</p> <ul style="list-style-type: none"> <li>• Informatiksysteme</li> <li>• Informatik, Mensch und Gesellschaft</li> </ul> <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> <li>• Einzelrechner</li> <li>• Dateisystem</li> <li>• Internet</li> <li>• Einsatz von Informatiksystemen</li> </ul> <p>Zeitbedarf: 6 Stunden</p>	<p>Inhaltsfelder:</p> <ul style="list-style-type: none"> <li>• Daten und ihre Strukturierung</li> <li>• Formale Sprachen und Automaten</li> </ul> <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> <li>• Objekte und Klassen</li> </ul> <p>Zeitbedarf: 8 Stunden</p>
---	---

Unterrichtsvorhaben EF-3	Unterrichtsvorhaben EF-4
<p>Thema: Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen in Java anhand von problemorientierten Szenarien</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> <li>• Argumentieren</li> <li>• Modellieren</li> <li>• Implementieren</li> <li>• Kommunizieren und Kooperieren</li> </ul> <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> <li>• Daten und ihre Strukturierung</li> <li>• Algorithmen</li> <li>• Formale Sprachen und Automaten</li> </ul> <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> <li>• Objekte und Klassen</li> <li>• Syntax und Semantik einer Programmiersprache</li> <li>• Analyse, Entwurf und Implementierung einfacher Algorithmen</li> </ul> <p>Zeitbedarf: 18 Stunden</p>	<p>Thema: Such- und Sortieralgorithmen anhand kontextbezogener Beispiele</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> <li>• Argumentieren</li> <li>• Modellieren</li> <li>• Darstellen und Interpretieren</li> <li>• Kommunizieren und Kooperieren</li> </ul> <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> <li>• Algorithmen</li> </ul> <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> <li>• Algorithmen zum Suchen und Sortieren</li> <li>• Analyse, Entwurf und Implementierung einfacher Algorithmen</li> </ul> <p>Zeitbedarf: 8 Stunden</p>

Unterrichtsvorhaben EF-5	Unterrichtsvorhaben EF-6
<p>Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen mit dem Schwerpunkt Assoziation anhand von einfachen Spielen</p>	<p>Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen mit dem Schwerpunkt Vererbung anhand von problemorientierten Szenarien</p>

<p>Zentrale Kompetenzen:</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>Inhaltsfelder:</p> <ul style="list-style-type: none"> <li>• Daten und ihre Strukturierung</li> <li>• Algorithmen</li> <li>• Formale Sprachen und Automaten</li> </ul> <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> <li>• Objekte und Klassen</li> <li>• Syntax und Semantik einer Programmiersprache</li> <li>• Analyse, Entwurf und Implementierung einfacher Algorithmen</li> </ul> <p>Zeitbedarf: 15 Stunden</p>	<p>Zentrale Kompetenzen:</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>Inhaltsfelder:</p> <ul style="list-style-type: none"> <li>• Daten und ihre Strukturierung</li> <li>• Algorithmen</li> <li>• Formale Sprachen und Automaten</li> </ul> <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> <li>• Objekte und Klassen</li> <li>• Syntax und Semantik einer Programmiersprache</li> <li>• Analyse, Entwurf und Implementierung einfacher Algorithmen</li> </ul> <p>Zeitbedarf: 15 Stunden</p>
---	---

<p><b>Unterrichtsvorhaben EF-7</b></p>
<p>Thema: Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> <li>• Argumentieren</li> <li>• Darstellen und Interpretieren</li> <li>• Kommunizieren und Kooperieren</li> </ul> <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> <li>• Informatik, Mensch und Gesellschaft</li> <li>• Informatiksysteme</li> </ul> <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> <li>• Wirkungen der Automatisierung</li> <li>• Geschichte der automatischen Datenverarbeitung</li> <li>• Digitalisierung</li> </ul> <p>Zeitbedarf: 18 Stunden</p>

**2.2.3 Qualifikationsphase – Q1**

<p>Unterrichtsvorhaben Q1-1</p>	<p>Unterrichtsvorhaben Q1-2</p>
---------------------------------	---------------------------------

<p>Thema: Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung</p> <p>Zentrale Kompetenzen:</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>Inhaltsfelder:</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>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> <li>• Objekte und Klassen</li> <li>• Analyse, Entwurf und Implementierung einfacher Algorithmen</li> <li>• Syntax und Semantik einer Programmiersprache</li> <li>• Nutzung von Informatiksystemen</li> </ul> <p>Zeitbedarf: 8 Stunden</p>	<p>Thema: Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen</p> <p>Zentrale Kompetenzen:</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>Inhaltsfelder:</p> <ul style="list-style-type: none"> <li>• Daten und ihre Strukturierung</li> <li>• Algorithmen</li> <li>• Formale Sprachen und Automaten</li> </ul> <p>Inhaltliche Schwerpunkte:</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>Zeitbedarf: 20 Stunden</p>
---	--

<p>Unterrichtsvorhaben Q1-3</p> <p>Thema: Suchen und Sortieren auf linearen Datenstrukturen</p> <p>Zentrale Kompetenzen:</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>Inhaltsfelder:</p> <ul style="list-style-type: none"> <li>• Algorithmen</li> <li>• Formale Sprachen und Automaten</li> </ul>	<p>Unterrichtsvorhaben Q1-4</p> <p>Thema: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten</p> <p>Zentrale Kompetenzen:</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>Inhaltsfelder:</p> <ul style="list-style-type: none"> <li>• Daten und ihre Strukturierung</li> <li>• Algorithmen</li> <li>• Formale Sprachen und Automaten</li> <li>• Informatik, Mensch und Gesellschaft</li> </ul>
---	--

<p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> <li>• Analyse, Entwurf und Implementierung einfacher Algorithmen</li> <li>• Algorithmen in ausgewählten informati- schen Kontexten</li> <li>• Syntax und Semantik einer Programmier- sprache</li> </ul> <p>Zeitbedarf: 16 Stunden</p>	<p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> <li>• Datenbanken</li> <li>• Algorithmen in ausgewählten informati- schen Kontexten</li> <li>• Syntax und Semantik einer Programmier- sprache</li> <li>• Sicherheit</li> </ul> <p>Zeitbedarf: 20 Stunden</p>
---	--

<p><b>Unterrichtsvorhaben Q1-5</b></p>
<p>Thema: Sicherheit und Datenschutz in Netz- strukturen</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> <li>• Argumentieren</li> <li>• Darstellen und Interpretieren</li> <li>• Kommunizieren und Kooperieren</li> </ul> <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> <li>• Informatiksysteme</li> <li>• Informatik, Mensch und Gesellschaft</li> </ul> <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> <li>• Einzelrechner und Rechnernetzwerke</li> <li>• Sicherheit</li> <li>• Nutzung von Informatiksystemen, Wir- kungen der Automatisierung</li> </ul> <p>Zeitbedarf: 10 Stunden</p>

**2.2.4 Qualifikationsphase – Q2**

<p><b>Unterrichtsvorhaben Q2-1</b></p>	<p><b>Unterrichtsvorhaben Q2-2</b></p>
<p>Thema: Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen</p> <p>Zentrale Kompetenzen:</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>Thema: Endliche Automaten und formale Spra- chen</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> <li>• Argumentieren</li> <li>• Modellieren</li> <li>• Darstellen und Interpretieren</li> <li>• Kommunizieren und Kooperieren</li> </ul>

<p>Inhaltsfelder:</p> <ul style="list-style-type: none"> <li>• Daten und ihre Strukturierung</li> <li>• Algorithmen</li> <li>• Formale Sprachen und Automaten</li> </ul> <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> <li>• Objekte und Klassen</li> <li>• Analyse, Entwurf und Implementierung einfacher Algorithmen</li> <li>• Algorithmen in ausgewählten informati-schen Kontexten</li> <li>• Syntax und Semantik einer Programmier-sprache</li> </ul> <p>Zeitbedarf: 24 Stunden</p>	<p>Inhaltsfelder:</p> <ul style="list-style-type: none"> <li>• Endliche Automaten und formale Spra-chen</li> </ul> <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> <li>• Endliche Automaten</li> <li>• Grammatiken regulärer Sprachen</li> <li>• Möglichkeiten und Grenzen von Automa-ten und formalen Sprachen</li> </ul> <p>Zeitbedarf: 20 Stunden</p>
---	--

<p><b>Unterrichtsvorhaben Q2-3</b></p>
<p>Thema: Prinzipielle Arbeitsweise eines Compu- ters und Grenzen der Automatisierbarkeit</p>
<p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> <li>• Argumentieren</li> <li>• Kommunizieren und Kooperieren</li> </ul>
<p>Inhaltsfelder:</p> <ul style="list-style-type: none"> <li>• Informatiksysteme</li> <li>• Informatik, Mensch und Gesellschaft</li> </ul>
<p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> <li>• Einzelrechner und Rechnernetzwerke</li> <li>• Grenzen der Automatisierung</li> </ul>
<p>Zeitbedarf: 12 Stunden</p>

## 2.3 Konkretisierte Unterrichtsvorhaben

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

Die Schülerinnen und Schüler

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

### Unterrichtsvorhaben Diff-8-1

*Thema:* Grundbegriffe der Informatik und Datenverarbeitung

*Leitfragen:* Wie können Informationen mithilfe eines Informatiksystems gespeichert und abgerufen werden?

*Zeitbedarf:* 20 Stunden

Das erste Unterrichtsvorhaben stellt eine allgemeine Einführung in das Fach Informatik dar. Die Schülerinnen und Schüler hatten vorher noch keinen Informatik-Unterricht.

Zunächst wird auf den Begriff der Information eingegangen und die Möglichkeit der Kodierung in Form von Daten thematisiert. Anschließend wird auf die Übertragung von Daten im Sinne des Sender-Empfänger-Modells eingegangen. Es werden zudem Sicherheitsfragen und Konzepte wie das Paritätsbit angesprochen.

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

### Unterrichtsvorhaben Diff-8-2

*Thema:* Einführung Algorithmen anhand der Entwicklung einfacher Spiele mit Scratch.

*Leitfragen:* Wie kann das Verhalten von Objekten situationsgemäß gesteuert und kontrolliert werden?

*Zeitbedarf:* 20 Stunden

Im Rahmen dieses Unterrichtsvorhabens wird der Objektbegriff kurz thematisiert und der Schwerpunkt auf das Verhalten bzw. Methoden von Objekten gelegt.

Die Schülerinnen und Schüler benutzen die Entwicklungsumgebung Scratch und lernen im Kontext von einfachen Spielen die zentralen algorithmischen Bausteine wie Verzweigung und Schleife kennen. Zudem werden Ereignisse und Nachrichten im Zuge der Objekt-Kommunikation thematisiert.

Dieses Unterrichtsvorhaben wird mit einem Projekt abgeschlossen.

**Unterrichtsvorhaben Diff-8-3**

*Thema:* Sichere Kommunikation in Rechnernetzen und insbesondere dem Internet

*Leitfragen:* Wie kann sicher über das Internet kommuniziert werden?

*Zeitbedarf:* 20 Stunden

Im ersten Block des Unterrichtsvorhabens wird zunächst die grundlegende Funktionsweise von Rechnernetzen und insbesondere dem Internet erarbeitet. Daran schließt sich die Betrachtung von möglichen Gefahren und Risiken der Internetkommunikation an. Für dieses Problem sollen informatische Lösungsansätze aus dem Bereich der Kryptologie betrachtet und reflektiert werden.

**Unterrichtsvorhaben Diff-8-4**

*Thema:* Entwicklung und Gestaltung von Webseiten

*Leitfragen:* Wie kann eine strukturierte und optisch ansprechende Webseite entwickelt werden?

*Zeitbedarf:* 20 Stunden

Zu Beginn dieses Unterrichtsvorhabens wird die Planung einer Webseite thematisiert. Anschließend wird die Planung mit HTML und CSS umgesetzt. Es werden zunächst die grundlegenden HTML-Tags eingeführt und anschließend zur Formatierung und verbesserten optischen Gestaltung der Webseite mit CSS gearbeitet. Die Schülerinnen und Schüler bekommen nur die grundlegenden Elemente vorgestellt und sollen durch Internet-Recherche weitere Elemente von HTML oder CSS finden und selbst erarbeiten.

Den Abschluss dieses Vorhabens bildet die Erstellung einer eigenen Webseite zu einem frei gewählten Thema im Rahmen eines Projektes.

**Unterrichtsvorhaben Diff-9-1**

*Thema:* Modellierung, Entwicklung von Datenbanken sowie deren Benutzung

*Leitfragen:* Wie kann man eine Datenbank entwerfen, umsetzen und damit arbeiten?

*Zeitbedarf:* 25 Stunden

Zunächst wird der Entwurf von Datenbanken mithilfe von ERM-Diagrammen thematisiert. Daran schließt sich die Überführung von ERM-Diagrammen in Tabellen bzw. Relationen an. Der Relationsbegriff wird hierbei nur kurz angesprochen und eher mit dem Tabellen-Begriff gearbeitet. Die konkrete Umsetzung der Tabellen kann mit LibreOffice Base oder SQLite (Werkzeug: SQLiteMan) erfolgen. Anschließend wird die Datenbanksprache SQL eingeführt und mit vorgegebenen SQLite-Datenbanken benutzt.

**Unterrichtsvorhaben Diff-9-2**

*Thema:* Datenbank-Projekt

*Leitfragen:* Wie kann für einen komplexen Kontext eine Datenbank modelliert und entwickelt werden?

*Zeitbedarf:* 15 Stunden

Die Schülerinnen und Schüler beschäftigen sich mit einem komplexen Inhaltsgebiet, für das eine Datenbank entwickelt werden soll. Dafür müssen zunächst Anforderungen gesammelt werden und ein ERM-Modell entwickelt werden. Die Überführung in Tabellen sowie die Umsetzung mit LibreOffice Base oder SQLite schließen sich an. Es muss zudem überlegt werden, welche Operationen ein Benutzer dieser Datenbank erwartet und benötigt. Diese werden in Form von SQL-Statements entwickelt.

Alle Phasen und Zwischenprodukte werden von den Schülerinnen und Schülern dokumentiert und in Form einer Projektmappe abgegeben.

### **Unterrichtsvorhaben Diff-9-3**

*Thema:* Grundlagen Objekte und Algorithmik

*Leitfragen:* Wie kann man Objekte beschreiben und wie kann man das Verhalten von Objekten modellieren?

*Zeitbedarf:* 20 Stunden

Im Rahmen dieses Unterrichtsvorhabens werden die Objektbegriff-Kenntnisse vertieft. Objekte werden in Form von Objektkarten beschrieben. Die Schülerinnen und Schüler wiederholen die zentralen algorithmischen Grundelemente wie Verzweigung und Schleife und stellen diese nun in Form von Struktogrammen dar. Ausgewählte Probleme werden nun betrachtet und algorithmisch gelöst.

### **Unterrichtsvorhaben Diff-9-4**

*Thema:* Einführung in objektorientiertes Programmieren

*Leitfragen:* Wie kann man Objekte mit einer textuellen Programmiersprache steuern und kontrollieren um Probleme zu lösen?

*Zeitbedarf:* 20 Stunden

Im Rahmen dieses Unterrichtsvorhabens wird zum ersten Mal mit einer textuellen Programmiersprache gearbeitet. Die Programmiersprache muss dynamische Typisierung erlauben, da so auf der Objektebene gearbeitet werden kann und der Klassenbegriff ausgeblendet werden kann. Mögliche Programmiersprachen sind Groovy, Boo oder Phyton. Aktuell wird Python erprobt.

Es wird grundsätzlich mit fertigen Objekten gearbeitet (Sprites sowie Maus und Tastatur aus der EgPyGame-Lib). Zunächst lernen die Schülerinnen und Schüler wie man diese Objekte mithilfe der Punktnotation steuern kann. Danach werden Variablen sowie die grundlegenden algorithmischen Bausteine thematisiert. Zum Abschluss der Reihe entwickeln die Schülerinnen und Schüler ein kleineres Spiel als Projektarbeit.

### **Unterrichtsvorhaben EF-1**

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

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

*Zeitbedarf:* 6 Stunden

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

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

Anhand der Untersuchung von normalen Desktop-Rechnern soll das EVA-Prinzip der Datenverarbeitung und das grundlegende Prinzip der Von-Neumann-Architektur erarbeitet werden.

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

<p>Sequenz 1: Information, deren Kodierung und Speicherung</p>	<p>Beispiele, Medien und Materialien</p>
<ul style="list-style-type: none"> <li>• Informatik als Wissenschaft der Verarbeitung von Informationen</li> <li>• Darstellung von Informationen in Schrift, Bild und Ton</li> <li>• Speichern von Daten mit informatischen Systemen am Beispiel der Schulrechner</li> <li>• Vereinbarung von Richtlinien zur Datenspeicherung auf den Schulrechnern (z.B. Ordnerstruktur, Dateibezeichner usw.)</li> </ul>	<p>Beispiel: Textkodierung</p> <p>Kodierung und Dekodierung von Texten anhand des ASCII-Codes und dem Binärsystem</p>
<p>Sequenz 2: Informations- und Datenübermittlung in Netzen</p>	<p>Beispiele, Medien und Materialien</p>
<ul style="list-style-type: none"> <li>• „Sender-Empfänger-Modell“ und seine Bedeutung für die Eindeutigkeit von Kommunikation</li> <li>• Informatische Kommunikation in Rechnernetzen am Beispiel des Schulnetzwerkes (z.B. Benutzeranmeldung, Netzwerkkordner, Zugriffsrechte, Client-Server)</li> <li>• Grundlagen der technischen Umsetzung von Rechnerkommunikation am Beispiel des Internets (z.B. Netzwerkadresse, Paketvermittlung, Protokoll, Datenpakete)</li> <li>• Richtlinien zum verantwortungsvollen Umgang mit dem Internet</li> </ul>	<p>Beispiel: Protokoll zur Anmeldung am Schulserver</p>
<p>Sequenz 3: Aufbau informatischer Systeme</p>	<p>Beispiele, Medien und Materialien</p>
<ul style="list-style-type: none"> <li>• Identifikation typischer Komponenten informatischer Systeme</li> <li>• Identifikation des EVA-Prinzips (Eingabe-Verarbeitung-Ausgabe) als Prinzip der Verarbeitung von Daten und Grundlage der „Von-Neumann-Architektur“</li> </ul>	<ul style="list-style-type: none"> <li>• Videos zu Hardware-Komponenten oder</li> <li>• Live-Vorführung einzelner Komponenten</li> </ul>
<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• beschreiben und erläutern den Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A),</li> <li>• nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D),</li> <li>• nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K).</li> </ul>	

**Unterrichtsvorhaben EF-2**

*Thema:* Grundlagen der objektorientierten Analyse und Modellierung

*Leitfrage:* Wie lassen sich Problembereiche informatisch modellieren?

*Zeitbedarf:* 8 Stunden

Ein zentraler Bestandteil des Informatikunterrichts der Einführungsphase ist die Objektorientierte Modellierung und Programmierung. Dieses Unterrichtsvorhaben führt schwerpunktmäßig in die Grundlagen der Analyse und Modellierung in diesem Kontext ein. Es werden zudem erste Grundsteine für die Implementierung durch z.B. die Punktnotation gelegt.

Dazu werden zunächst konkrete Gegenstandsbereiche aus der Lebenswelt der Schülerinnen und Schüler analysiert und im Sinne des Objektorientierten Paradigmas strukturiert. Das Abstraktionsniveau bleibt dabei zunächst auf der Objektebene. Es werden die grundlegenden Begriffe der Objektorientierung und Modellierungswerkzeuge wie Objektkarten, Objektdiagramme und Sequenzdiagramme eingeführt.

Sequenz 1: Identifikation von Objekten	Beispiele, Medien und Materialien
<ul style="list-style-type: none"> <li>• Am Beispiel eines lebensweltnahen Beispiels werden Objekte im Sinne der Objektorientierten Modellierung eingeführt.</li> <li>• Objekte werden mit Objektkarten visualisiert und mit sinnvollen Attributen und „Fähigkeiten“, d.h. Methoden versehen.</li> <li>• Methoden werden unterteilt in Aufträge und Anfragen.</li> <li>• Aufruf von Methoden durch Punktnotation.</li> <li>• Einführung von Parametern als Zusatzinformationen für eine Methode.</li> <li>• Unterschied zwischen formalen Parametern bei der Methodendefinition und Parameterwerten beim Methodenaufruf.</li> <li>• Anwendung und Überleitung zum nächsten Unterrichtsvorhaben: Analyse des Hamstermodells aus objektorientierter Sicht.</li> </ul>	<ul style="list-style-type: none"> <li>• Smartphone, PC o.ä. Einstiegsbeispiel für Objekte</li> </ul>
<p>Zu entwickelnde Kompetenzen:</p>	
<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>• stellen die Kommunikation zwischen Objekten grafisch dar (M),</li> <li>• stellen den Zustand eines Objekts dar (D).</li> <li>• analysieren und erläutern eine objektorientierte Modellierung (A)</li> </ul>	

**Unterrichtsvorhaben EF-3**

*Thema:* Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen in Java anhand von problemorientierten Szenarien

*Leitfragen:* Wie kann man Probleme durch algorithmische Elemente lösen?

*Zeitbedarf:* 18 Stunden

Der Schwerpunkt dieses Unterrichtsvorhabens liegt in der Bearbeitung von verschiedenen kleineren Problemszenarien innerhalb des Hamstermodells. Der Hamster wird als ein Objekt betrachtet, das verschiedene Methoden und Attribute besitzt. Mithilfe dieser bereits vorhandenen Elemente sollen die Problemszenarien bearbeitet werden, was eine Erweiterung der Methoden und Attribute des Hamsters erfordert. Hier werden die Grundlagen für das Implementieren von Methoden und Attributen gelegt. Im Rahmen dieser Erweiterung werden Kontrollstrukturen innerhalb der Methoden benötigt, die hier erstmalig eingeführt werden. Für die Realisierung von Attributen wird das Variablenkonzept eingeführt und später auf Parameter und lokale Variablen übertragen. Im Rahmen dieser objekt-orientierten Algorithmik werden Algorithmen durch Struktogramme modelliert und dargestellt. Die Umsetzung in Java schließt sich daran an. Integrativ werden kleinere mathematische Berechnungen beim Bereich der Wertveränderung von Variablen bzw. Attributen behandelt. Im Bereich der Algorithmik wird das Thema Rekursion nicht behandelt (später in der Q1). Beziehungen zwischen Objekten werden in diesem Unterrichtsvorhaben zunächst nicht behandelt. Sie stellen den Schwerpunkt des Unterrichtsvorhabens EF-05 dar.

Sequenz 1: Einführung Anweisungen und Methoden	Beispiele, Medien und Materialien
<ul style="list-style-type: none"> <li>• Aufruf von einfachen Methoden innerhalb einer Methode durch Punktnotation (Schlüsselwort „this“)</li> <li>• Anlegen von eigenen Methoden: Aufträge.</li> </ul>	Sammlung Hamster-Aufgaben
Sequenz 2: Attribute und Variablenkonzept	Beispiele, Medien und Materialien
<ul style="list-style-type: none"> <li>• Variablen zur Umsetzung von Attributen</li> <li>• elementare Datentypen</li> <li>• Deklaration, Wertzuweisung, Wertveränderung, Benutzung</li> <li>• Unterschied lokale Variablen und globale Variablen (Attribute) hinsichtlich Sichtbarkeit bzw. Benutzbarkeit</li> </ul>	Sammlung Hamster-Aufgaben
Sequenz 3: Verzweigung	Beispiele, Medien und Materialien
<ul style="list-style-type: none"> <li>• Verzweigung und bedingte Anweisungen</li> <li>• Bedingungen</li> <li>• Darstellung im Struktogramm und Umsetzung mit if-else in Java.</li> <li>• Mehrfachauswahl mit if-else if-else</li> </ul>	Sammlung Hamster-Aufgaben
Sequenz 4: Erweiterung der Methoden um Anfragen	Beispiele, Medien und Materialien

<ul style="list-style-type: none"> <li>• Methoden mit einer Rückgabe in Java</li> <li>• Benutzung von Anfragen, insbesondere von Boolean-Anfragen in Bedingungen</li> </ul>	Sammlung Hamster-Aufgaben
Sequenz 5: Vertiefung von Bedingungen durch logische Verknüpfungen	Beispiele, Medien und Materialien
<ul style="list-style-type: none"> <li>• Bedingung kann aus mehreren Bedingungen bestehen, die miteinander verknüpft sind</li> <li>• Logische Verknüpfungen Und, Oder, Negation</li> <li>• Vertiefung Umgang mit Booleans</li> </ul>	Sammlung Hamster-Aufgaben
Sequenz 6: Wiederholungen	Beispiele, Medien und Materialien
<ul style="list-style-type: none"> <li>• Die While-Schleife im Struktogramm und in Java</li> <li>• Die Do-While-Schleife</li> <li>• Die Zählschleife for</li> </ul>	Sammlung Hamster-Aufgaben
Sequenz 7: Methoden mit Parametern	Beispiele, Medien und Materialien
<ul style="list-style-type: none"> <li>• Parameter als lokale Variablen</li> <li>• Anlegen von Methoden mit Parametern und Benutzung des Parameters</li> <li>• Aufruf von Methoden mit Parametern</li> </ul>	Sammlung Hamster-Aufgaben
Sequenz 8: Strategien zum Lösen von komplexen Problemen	Beispiele, Medien und Materialien
<ul style="list-style-type: none"> <li>• Zentrale Strategie: Problemzerlegung</li> <li>• Top-Down-Prinzip</li> <li>• Bottom-Up-Prinzip</li> </ul>	Sammlung Hamster-Aufgaben
Zu entwickelnde Kompetenzen:	

Die Schülerinnen und Schüler

- analysieren und erläutern einfache Algorithmen und Programme (A), entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M),
- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M),
- modifizieren einfache Algorithmen und Programme (I),
- implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I),
- implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),
- testen Programme schrittweise anhand von Beispielen (I),
- interpretieren Fehlermeldungen und korrigieren den Quellcode (I).

**Unterrichtsvorhaben EF-4**

*Thema:* Such- und Sortieralgorithmen anhand kontextbezogener Beispiele

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

*Zeitbedarf:* 8 Stunden

Dieses Unterrichtsvorhaben beschäftigt sich mit der Erarbeitung von Such- und Sortieralgorithmen. Der Schwerpunkt des Vorhabens liegt dabei auf den Algorithmen selbst und nicht auf deren Implementierung in einer Programmiersprache, auf die in diesem Vorhaben vollständig verzichtet werden soll. Der Einstieg in das Gebiet folgt über die zentrale Aufgabe eines Informatiksystems: die Suche. Es wird neben der linearen Suche die binäre Suche thematisiert und anhand dieser, die große Bedeutung von sortieren Daten herausgestellt. Im Bereich des Sortierens wird mit den Schülern der einfache und schlechte Sortieralgorithmus Sortieren durch Suchen besprochen, der aber sehr intuitiv ist. Anhand dieses Beispielalgorithmus, bei dem man einfach alle Zahlen von 1 bis 1000 sucht, wird in den Bereich der Suchstrategien (Einsatzszenarien, Effizienz, Korrektheit) eingeführt. Nach diesem Beispiel erarbeiten bzw. entwickeln die SuS arbeitsteilig ein weiteres Verfahren anhand von Selbstlernmaterialien und präsentieren dies der Klasse. Die Schülerinnen und Schüler sollen auf diese Weise das Sortieren durch Vertauschen („Bubblesort“), das Sortieren durch Auswählen („Selectionsort“) das Sortieren durch Mischen („Mergesort“) kennen lernen.

Sequenz 1: Suchen	Beispiele, Medien und Materialien
<ul style="list-style-type: none"> <li>• Suchen als zentrale Aufgabe eines Informationssystems.</li> <li>• Lineare Suche und Analyse ihrer Geschwindigkeit.</li> <li>• Binäre Suche und Analyse der Geschwindigkeit</li> <li>• Herausstellen der großen Bedeutung einer sortierten Datenmenge.</li> </ul>	Lange Namensliste (sortiert und unsortiert) als Einstiegsbeispiel
Sequenz 2: Sortieren	Beispiele, Medien und Materialien

<ul style="list-style-type: none"> <li>• Einstiegsalgorithmus: Sortieren durch Suchen. Analyse von</li> <li>• Geschwindigkeit, Korrektheit und Effizienz.</li> <li>• Erarbeitung von zwei Algorithmen (Bubblesort, Selectionsort oder Mergesort).</li> <li>• Vergleichende Analyse der Algorithmen</li> </ul>	<ul style="list-style-type: none"> <li>• Videos zur Visualisierung der Algorithmen.</li> <li>• Festlegung: primär Sortieren von Zahlen</li> </ul>
Zu entwickelnde Kompetenzen:	
Die Schülerinnen und Schüler <ul style="list-style-type: none"> <li>• beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeit und Speicherplatzbedarf (A),</li> <li>• entwerfen einen weiteren Algorithmus zum Sortieren (M),</li> <li>• analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D).</li> </ul>	

**Unterrichtsvorhaben EF-5**

*Thema:* Modellierung und Implementierung von Klassen- und Objektbeziehungen mit dem Schwerpunkt Assoziation anhand von einfachen Spielen

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

*Zeitbedarf:* 15 Stunden

Dieses Unterrichtsvorhaben beschäftigt sich im Schwerpunkt mit dem Aufbau von Objektbeziehungen durch Assoziationen. Über Beziehungsattribute können Objekte eine Beziehung zu einem anderen Objekt eingehen und dessen Dienste nutzen. Dazu bedarf es zunächst einer präzisen Unterscheidung zwischen Objektreferenzen und Objekten, so dass klar wird, dass Dienste eines Objektes von unterschiedlichen Objekten über unterschiedliche Referenzen in Anspruch genommen werden können. Auch der Aufbau solcher Objektbeziehungen (inklusive Verwaltungsmethoden) muss thematisiert werden. Zur Anwendung und Vertiefung kann ggf. eine grafische Oberfläche erstellt werden, die zunächst nur über Assoziationen bzw. Beziehungsattribute realisiert wird. Die Fokussierung liegt aber auf der eigentlichen inhaltlichen Modellierung und Implementierung.

Sequenz 1: Wiederholungen objektorientierte Analyse und Modellierung	Beispiele, Medien und Materialien
<ul style="list-style-type: none"> <li>• Identifikation von Objekten und Modellierung ihrer Attribute und Methoden mit dem Abbott-Verfahren.</li> <li>• Analyse und Modellierung der Beziehungen zwischen Objekten, die durch Beziehungsattribute realisiert werden. Darstellung durch Objektdiagramme.</li> <li>• Vertiefung der Verwaltungsmethoden für Beziehungen.</li> <li>• Modellierung der Methodeninhalte auf Objektebene durch Strukturgramme.</li> </ul>	Werkzeuge: <ul style="list-style-type: none"> <li>• Objektdiagramm</li> <li>• Sequenzdiagramm</li> </ul> Mögliche Beispiele: <ul style="list-style-type: none"> <li>• Nimm-Spiel</li> <li>• Verflixte Eins</li> </ul>

<p>Sequenz 2: Einführung des Klassenbegriffs und Abstraktion der Objekte zu Klassen</p>	<p>Beispiele, Medien und Materialien</p>
<ul style="list-style-type: none"> <li>• Klassenbegriff als Bauplan</li> <li>• Klassen als eigene Datentypen</li> <li>• Darstellung von Klassen durch Klassenkarte</li> <li>• Abstraktion des Objektdiagramms zum Klassendiagramm</li> <li>• Konstruktor (Standardkonstruktor vs. erweiterter Konstruktor mit Parametern)</li> </ul>	<p>Werkzeuge:</p> <ul style="list-style-type: none"> <li>• Objektdiagramm</li> <li>• Sequenzdiagramm</li> </ul> <p>Mögliche Beispiele:</p> <ul style="list-style-type: none"> <li>• Nimm-Spiel</li> <li>• Verflixte Eins</li> </ul>
<p>Sequenz 3: Implementierung eines objektorientierten Modells mit Java</p>	<p>Beispiele, Medien und Materialien</p>
<ul style="list-style-type: none"> <li>• Umsetzung Klassengerüst</li> <li>• Umsetzung Konstruktor</li> <li>• Umsetzung Methodeninhalte</li> <li>• Ggf. Main-Methode und Dokumentation</li> </ul>	<p>Werkzeuge:</p> <ul style="list-style-type: none"> <li>• Objektdiagramm</li> <li>• Sequenzdiagramm</li> </ul> <p>Mögliche Beispiele:</p> <ul style="list-style-type: none"> <li>• Nimm-Spiel</li> <li>• Verflixte Eins</li> </ul>
<p>Zu entwickelnde Kompetenzen:</p>	
<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>• stellen die Kommunikation zwischen Objekten grafisch dar (M),</li> <li>• stellen den Zustand eines Objekts dar (D).</li> <li>• analysieren und erläutern eine objektorientierte Modellierung (A)</li> <li>• modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M),</li> <li>• implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M),</li> <li>• ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),</li> <li>• testen Programme schrittweise anhand von Beispielen (I),</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>• modifizieren einfache Algorithmen und Programme (I),</li> <li>• dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D).</li> </ul>	

**Unterrichtsvorhaben EF-6**

*Thema:* Modellierung und Implementierung von Klassen- und Objektbeziehungen mit dem Schwerpunkt Vererbung anhand von problemorientierten Szenarien

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

*Zeitbedarf:* 15 Stunden

Dieses Unterrichtsvorhaben beschäftigt sich im Schwerpunkt mit dem Aufbau komplexerer Objektbeziehungen. Während in vorangegangenen Unterrichtsvorhaben Objekte nur mit anderen Objekten über Beziehungsattribute kommuniziert haben und dies über Assoziationen auf Klassenebene umgesetzt wurde, sollen nun komplexere Beziehungen durch Einführung der Vererbung als weiteren Abstraktionsschritt ermöglicht werden. Die Vererbung soll als Abstraktionsschritt eingeführt werden. Daher wird ein Beispiel zunächst normal modelliert und anhand von gemeinsamen Methoden und Attributen soll die Abstraktion einer Oberklasse mit spezialisierten Unterklassen realisiert werden. Durch einige exemplarische Problemstellungen werden die wichtigsten Varianten der Vererbung vorgestellt. Zum Abschluss wird kurz auf das Prinzip der abstrakten Klasse eingegangen werden.

Sequenz 1: Einführung der Vererbung auf Modellebene	Beispiele, Medien und Materialien
<ul style="list-style-type: none"> <li>• Einführung der Vererbung durch Herausstellung von gemeinsamen Attributen und Methoden von ähnlichen Klassen.</li> <li>• Vererbung als ist-Beziehung</li> <li>• Ober- und Unterklassen, Vererbungsprinzip, Generalisierung und Spezialisierung</li> <li>• Spezialisierung von Methoden bzw. Überschreiben von Methoden</li> <li>• Benutzung von Vererbung durch abstraktere Beziehungsattribute</li> </ul>	Mögliche Beispiele: <ul style="list-style-type: none"> <li>• Zoo-Verwaltung</li> <li>• Media-Player</li> <li>• Zugverwaltung</li> </ul>
Sequenz 2: Implementierung der Vererbung in Java	Beispiele, Medien und Materialien
<ul style="list-style-type: none"> <li>• Umsetzung der Vererbungsbeziehung in Java durch das Schlüsselwort „extends“</li> <li>• Umsetzung des Konstruktors und Einführung des Schlüsselwortes „super“</li> <li>• Umgang mit Oberklassen-Datentypen z.B. bei Beziehungsattributen und Parametern.</li> </ul>	Mögliche Beispiele: <ul style="list-style-type: none"> <li>• Zoo-Verwaltung</li> <li>• Media-Player</li> <li>• Zugverwaltung</li> </ul>
Sequenz 3: Abstrakte Klassen	Beispiele, Medien und Materialien

<ul style="list-style-type: none"> <li>• Einführung Abstrakte Klassen auf Modellebene</li> <li>• Zwang der Spezialisierung von Methoden</li> <li>• Umsetzung von abstrakten Klassen in Java</li> <li>• Typumwandlung und damit verbundene Probleme</li> <li>• Sicherheit beim Typecast durch instanceof-Operator</li> </ul>	<p>Mögliche Beispiele:</p> <ul style="list-style-type: none"> <li>• Zoo-Verwaltung</li> <li>• Media-Player</li> <li>• Zugverwaltung</li> </ul>
<p>Zu entwickelnde Kompetenzen:</p>	
<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• objektorientierte Modellierung (A),</li> <li>• stellen die Kommunikation zwischen Objekten grafisch dar (M),</li> <li>• ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li> <li>• modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M),</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M),</li> <li>• ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),</li> <li>• modellieren Klassen unter Verwendung von Vererbung (M),</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),</li> <li>• testen Programme schrittweise anhand von Beispielen (I),</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>• modifizieren einfache Algorithmen und Programme (I),</li> <li>• stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D),</li> <li>• dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D)</li> </ul>	

**Unterrichtsvorhaben EF-7**

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

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

*Zeitbedarf:* 4 Stunden

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

Anschließend wird verstärkt auf den Aspekt des Datenschutzes eingegangen. Dazu wird das Bundesdatenschutzgesetz in Auszügen behandelt und auf schülernahe Beispielsituationen zur Anwendung

gebracht. Dabei steht keine formale juristische Bewertung der Beispielsituationen im Vordergrund, die im Rahmen eines Informatikunterrichts auch nicht geleistet werden kann, sondern vielmehr eine persönliche Einschätzung von Fällen im Geiste des Datenschutzgesetzes.

Die Referate können ggf. auch bei anderen Unterrichtsvorhaben eingebunden werden.

Sequenz 1: Datenverarbeitung und Datenschutz	Beispiele, Medien und Materialien
Selbstständige Erarbeitung von Themen durch die Schülerinnen und Schüler in Kleingruppen <ul style="list-style-type: none"> <li>• Kryptologie</li> <li>• Auswirkungen der Digitalisierung</li> <li>• Kodierung</li> <li>• Geschichte der Digitalisierung</li> <li>• Rechnen im Binärsystem</li> </ul> Erarbeitung grundlegender Begriffe des Datenschutzes Problematisierung und Anknüpfung an die Lebenswelt der Schülerinnen und Schüler Diskussion und Bewertung von Fallbeispielen aus dem Themenbereich „Datenschutz“	Ausgewählte Texte und selbstständige Recherche im Internet, in der Schulbibliothek, in öffentlichen Bibliotheken, usw.
Zu entwickelnde Kompetenzen:	
Die Schülerinnen und Schüler <ul style="list-style-type: none"> <li>• bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A),</li> <li>• erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A),</li> <li>• stellen ganze Zahlen und Zeichen in Binärcodes dar (D),</li> <li>• interpretieren Binärcodes als Zahlen und Zeichen (D),</li> <li>• nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation. (K).</li> </ul>	

**Konkretisierte Unterrichtsvorhaben Qualifikationsphase**

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

Die Schülerinnen und Schüler

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

**Unterrichtsvorhaben Q1-1**

*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: 8 Stunden

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

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

<p>Sequenz 1: Wiederholung und Erweiterung der objektorientierten Modellierung und Programmierung durch Analyse und Erweiterung eines kontextbezogenen Beispiels</p>	<p>Beispiele, Medien und Materialien</p>
<ul style="list-style-type: none"> <li>• Analyse der Problemstellung</li> <li>• Analyse der Modellierung (Implementationsdiagramm)</li> <li>• Erweiterung der Modellierung im Implementationsdiagramm (Vererbung, abstrakte Klasse)</li> <li>• Kommunikation zwischen mindestens zwei Objekten (grafische Darstellung)</li> <li>• Dokumentation von Klassen</li> <li>• Implementierung der Anwendung oder von Teilen der Anwendung</li> </ul>	<p>Medien- oder Zug-Projekt</p>
<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• analysieren und erläutern objektorientierte Modellierungen (A),</li> <li>• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (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> <li>• implementieren Klassen in einer Programmiersprache 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>• stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D),</li> <li>• dokumentieren Klassen (D),</li> <li>• stellen die Kommunikation zwischen Objekten grafisch dar (D).</li> </ul>	

**Unterrichtsvorhaben EF-3**

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

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

*Zeitbedarf:* 20 Stunden

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

notwendige Klassen modelliert und implementiert. Eine Klasse für eine den Anforderungen der Anwendung entsprechende Oberfläche sowie die Klasse Queue wird dabei von der Lehrkraft vorgegeben. Anschließend wird die Anwendung modifiziert, um den Umgang mit der Datenstruktur zu üben. Anhand einer Anwendung, in der Daten nach dem Last-In-First-Out-Prinzip verwaltet werden, werden Unterschiede zwischen den Datenstrukturen Schlange und Stapel erarbeitet. Um einfacher an Objekte zu gelangen, die zwischen anderen gespeichert sind, wird die Klasse List eingeführt und in einem Anwendungskontext verwendet. In mindestens einem weiteren Anwendungskontext wird die Verwaltung von Daten in Schlangen, Stapeln oder Listen vertieft. Modellierungen werden dabei in Entwurfs- und Implementationsdiagrammen dargestellt.

Sequenz 1: Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse Queue	Beispiele, Medien und Materialien
<ul style="list-style-type: none"> <li>• Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</li> <li>• Erarbeitung der Funktionalität der Klasse Queue</li> <li>• Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Queue</li> </ul>	Patientenwarteschlange
Sequenz 2: Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse Stack	Beispiele, Medien und Materialien
<ul style="list-style-type: none"> <li>• Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</li> <li>• Erarbeitung der Funktionalität der Klasse Stack</li> <li>• Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Stack</li> </ul>	Klausurstapel mit automatischer Benotung, Versicherungsprojekt, Zugaufteilungprojekt, Palindrom-Checker
Sequenz 3: Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse List	Beispiele, Medien und Materialien
<ul style="list-style-type: none"> <li>• Erarbeitung der Vorteile der Klasse List im Gegensatz zu den bereits bekannten linearen Strukturen</li> <li>• Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse List.</li> </ul>	Abfahrtslauf, Kontaktliste
Sequenz 4: Vertiefung linearer Datenstrukturen	Beispiele, Medien und Materialien

<p>Anwendungen von Listen, Stapeln oder Schlangen in mindestens einem weiteren Kontext</p> <ul style="list-style-type: none"> <li>• Wahl der Datenstruktur</li> <li>• Verwendung mehrerer Datenstrukturen</li> </ul>	<ul style="list-style-type: none"> <li>• Beispiel: Kontaktliste</li> <li>• Beispiel: Abfahrtslauf</li> <li>• Beispiel: Skispringen</li> <li>• Beispiel: Rangierbahnhof</li> <li>• Beispiel: Autos an einer Ampel zur Zufahrtsregelung</li> </ul>
<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A),</li> <li>• analysieren und erläutern Algorithmen und Programme (A),</li> <li>• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nicht-lineare Datensammlungen zu (M),</li> <li>• ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li> <li>• modifizieren Algorithmen und Programme (I),</li> <li>• implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),</li> <li>• nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>• testen Programme systematisch anhand von Beispielen (I),</li> <li>• stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D).</li> </ul>	

**Unterrichtsvorhaben Q1-3**

*Thema:* Suchen und Sortieren auf linearen Datenstrukturen

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

*Zeitbedarf:* 16 Stunden

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

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

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

<p>Sequenz 1: Suchen von Daten in Listen und Arrays</p>	<p>Beispiele, Medien und Materialien</p>
---	--

<ul style="list-style-type: none"> <li>• Lineare Suche in Listen und in Arrays</li> <li>• Binäre Suche in Arrays als Beispiel für rekursives Problemlösen</li> <li>• Untersuchung der beiden Suchverfahren hinsichtlich ihrer Effizienz (Laufzeitverhalten, Speicherbedarf)</li> </ul>	<p>Suchen in einem einfachen Zahlenfeld, Suchen in einer Kontaktkartei</p>
<p>Sequenz 2: Sortieren in Listen und Arrays - Entwicklung und Implementierung von iterativen und rekursiven Sortierverfahren</p>	<p>Beispiele, Medien und Materialien</p>
<ul style="list-style-type: none"> <li>• Entwicklung und Implementierung eines einfachen Sortierverfahrens für eine Liste</li> <li>• Implementierung eines einfachen Sortierverfahrens für ein Feld</li> <li>• Entwicklung eines rekursiven Sortierverfahrens für ein Feld (z.B. Sortieren durch Mischen)</li> </ul>	<p>Suchen in einem einfachen Zahlenfeld, Suchen in einer Kontaktkartei</p>
<p>Sequenz 3: Untersuchung der Effizienz der Sortierverfahren „SelectionSort“ und „Quicksort“ auf linearen Listen</p>	<p>Beispiele, Medien und Materialien</p>
<ul style="list-style-type: none"> <li>• Grafische Veranschaulichung der Sortierverfahren</li> <li>• Untersuchung der Anzahl der Vergleichsoperationen und des Speicherbedarf bei beiden Sortierverfahren</li> <li>• Beurteilung der Effizienz der beiden Sortierverfahren</li> </ul>	<p>Suchen in einem einfachen Zahlenfeld, Suchen in einer Kontaktkartei</p>

## Die Schülerinnen und Schüler

- analysieren und erläutern Algorithmen und Programme (A),
- beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),
- beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A),
- entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M),
- modifizieren Algorithmen und Programme (I),
- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),
- implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I),
- nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),
- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),
- testen Programme systematisch anhand von Beispielen (I),
- stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).

**Unterrichtsvorhaben Q1-4**

*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?

*Zeitbedarf:* 20 Stunden

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

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

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

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

Sequenz 1: Grundlagen und Aufbau von Datenbanken	
--	--

<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>	Video-Center, Musikshop, Buchausleihe, Schulverwaltung
Sequenz 2: Nutzung von relationalen Datenbanken mit SQL-Abfragen	Beispiele, Medien und Materialien
<ul style="list-style-type: none"> <li>• Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL (SELECT (DISTINCT) ...FROM, WHERE, AND, OR, NOT) auf einer Tabelle</li> <li>• Analyse und Erarbeitung von SQL-Abfragen auf einer und mehrerer Tabelle zur Beantwortung der Fragestellungen (JOIN, UNION, AS, GROUP BY,ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, &lt;&gt;, &gt;, &lt;, &gt;=, &lt;=, LIKE, BETWEEN, IN, IS NULL</li> </ul>	Video-Center, Musikshop, Buchausleihe, Schulverwaltung
Sequenz 3: Modellierung und Optimierung von relationalen Datenbanken	Beispiele, Medien und Materialien

<p>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>Entwicklung einer Datenbank aus einem Datenbankentwurf</p> <ul style="list-style-type: none"> <li>• Modellierung eines relationalen Datenbankschematas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln</li> </ul> <p>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>	<p>Video-Center, Musikshop, Buchausleihe, Schulverwaltung</p>
---	---

Zu entwickelnde Kompetenzen:

Die Schülerinnen und Schüler

- erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A),
- analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A),
- analysieren und erläutern eine Datenbankmodellierung (A),
- erläutern die Eigenschaften normalisierter Datenbankschemata (A),
- bestimmen Primär- und Sekundärschlüssel (M),
- ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M),
- modifizieren eine Datenbankmodellierung (M),
- modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M),
- bestimmen Primär- und Sekundärschlüssel (M),
- überführen Datenbankschemata in vorgegebene Normalformen (M),
- verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I),
- ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D),
- stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D),
- überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D).

**Unterrichtsvorhaben Q1-5**

*Thema:* Sicherheit und Datenschutz in Netzstrukturen

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

*Zeitbedarf:* 10 Stunden

Anschließend an das vorhergehende Unterrichtsvorhaben zum Thema Datenbanken werden der Datenbankzugriff aus dem Netz, Topologien von Netzwerken, eine Client-Server-Struktur, das TCP/IP-Schichtenmodell sowie Sicherheitsaspekte beim Zugriff auf Datenbanken und verschiedene symmetrische und asymmetrische kryptografische Verfahren analysiert und erläutert. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht runden das Unterrichtsvorhaben ab.

Sequenz 1: Daten in Netzwerken und Sicherheitsaspekte in Netzen sowie beim Zugriff auf Datenbanken	Beispiele, Medien und Materialien
<ul style="list-style-type: none"> <li>• Beschreibung eines Datenbankzugriffs im Netz anhand eines Anwendungskontextes und einer Client-Server-Struktur zur Klärung der Funktionsweise eines Datenbankzugriffs</li> <li>• Netztopologien als Grundlage von Client-Server-Strukturen und TCP/IP-Schichtenmodell als Beispiel für eine Paketübermittlung in einem Netz</li> <li>• Vertraulichkeit, Integrität, Authentizität in Netzwerken sowie symmetrische und asymmetrische kryptografische Verfahren (Cäsar-, Vigenère-, RSA-Verfahren) als Methoden Daten im Netz verschlüsselt zu übertragen</li> <li>• Optional: Vertiefung durch Implementierung einem Projekt mit Client-Server-Strukturen</li> </ul>	ToDo
Sequenz 2: Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht	Beispiele, Medien und Materialien
	ToDo
Zu entwickelnde Kompetenzen:	

Die Schülerinnen und Schüler

- beschreiben und erläutern Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A),
- analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A),
- 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),
- 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),
- nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D).

**Unterrichtsvorhaben Q2-1**

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

*Leitfrage:* 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

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

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

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

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

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

Sequenz 1: Analyse von Baumstrukturen in verschiedenen Kontexten

Beispiele, Medien und Materialien

<ul style="list-style-type: none"> <li>• Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit)</li> <li>• Aufbau und Darstellung von binären Bäumen anhand von Baumstrukturen in verschiedenen Kontexten</li> </ul>	<p>Mögliche Beispiele:</p> <ul style="list-style-type: none"> <li>• Ahnenbaum</li> <li>• Codierungsbäume wie z.B. Morsecode</li> <li>• Akinator</li> </ul>
<p>Sequenz 2: Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse BinaryTree</p>	<p>Beispiele, Medien und Materialien</p>
<ul style="list-style-type: none"> <li>• Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext</li> <li>• Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms</li> <li>• Erarbeitung der Klasse BinaryTree und beispielhafte Anwendung der Operationen</li> <li>• Implementierung der Anwendung oder von Teilen der Anwendung</li> <li>• Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf</li> </ul>	<p>Mögliche Beispiele:</p> <ul style="list-style-type: none"> <li>• Ahnenbaum</li> <li>• Codierungsbäume wie z.B. Morsecode</li> <li>• Akinator</li> </ul>
<p>Sequenz 3: Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse BinarySearchTree</p>	<p>Beispiele, Medien und Materialien</p>
<ul style="list-style-type: none"> <li>• Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</li> <li>• Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms, grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften</li> <li>• Erarbeitung der Klasse BinarySearchTree und Einführung des Interface Item zur Realisierung einer geeigneten Ordnungsrelation</li> <li>• Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums</li> </ul>	<p>Mögliche Beispiele:</p> <ul style="list-style-type: none"> <li>• Ahnenbaum</li> <li>• Codierungsbäume wie z.B. Morsecode</li> <li>• Akinator</li> </ul>
<p>Sequenz 4: Übung und Vertiefungen der nicht-linearen Datenstruktur Baum</p>	<p>Beispiele, Medien und Materialien</p>

Anwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen	Mögliche Beispiele: <ul style="list-style-type: none"> <li>• Ahnenbaum</li> <li>• Codierungsbäume wie z.B. Morsecode</li> <li>• Akinator</li> </ul>
Zu entwickelnde Kompetenzen:	
Die Schülerinnen und Schüler <ul style="list-style-type: none"> <li>• erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A),</li> <li>• analysieren und erläutern Algorithmen und Programme (A),</li> <li>• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li> <li>• ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M),</li> <li>• modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),</li> <li>• verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M),</li> <li>• entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M),</li> <li>• implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),</li> <li>• modifizieren Algorithmen und Programme (I),</li> <li>• nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>• testen Programme systematisch anhand von Beispielen (I),</li> <li>• stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D),</li> <li>• stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).</li> </ul>	

**Unterrichtsvorhaben Q2-2**

*Thema:* Endliche Automaten und formale Sprachen

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

*Zeitbedarf:* 20 Stunden

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

Anhand kontextbezogener Beispiele werden Grammatiken regulärer Sprachen entwickelt, untersucht und modifiziert. Der Zusammenhang zwischen regulären Grammatiken und endlichen Automaten wird verdeutlicht durch die Entwicklung von allgemeinen Verfahren zur Erstellung einer regulären

Grammatik für die Sprache eines gegebenen endlichen Automaten bzw. zur Entwicklung eines endlichen Automaten, der genau die Sprache einer gegebenen regulären Grammatik akzeptiert.

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

Sequenz 1: Endliche Automaten	Beispiele, Medien und Materialien
<ul style="list-style-type: none"> <li>• Vom Automaten in den Schülerinnen und Schülern bekannten Kontexten zur formalen Beschreibung eines endlichen Automaten</li> <li>• Untersuchung, Darstellung und Entwicklung endlicher Automaten</li> </ul>	Mögliche Beispiele: <ul style="list-style-type: none"> <li>• Cola-Automat</li> <li>• Parkautomat</li> <li>• Enigma</li> <li>• Mastermind</li> </ul>
Sequenz 2: Untersuchung und Entwicklung von Grammatiken regulärer Sprachen	
<ul style="list-style-type: none"> <li>• Erarbeitung der formalen Darstellung regulärer Grammatiken</li> <li>• Untersuchung, Modifikation und Entwicklung von Grammatiken</li> <li>• Entwicklung von endlichen Automaten zum Erkennen regulärer Sprachen die durch Grammatiken gegeben werden</li> <li>• Entwicklung regulärer Grammatiken zu endlichen Automaten</li> </ul>	reguläre Grammatik für Wörter mit ungerader Parität, Grammatik für Wörter, die bestimmte Zahlen repräsentieren, Satzgliederungsgrammatik
Sequenz 3: Grenzen endlicher Automaten	Beispiele, Medien und Materialien
	Mögliche Beispiele: <ul style="list-style-type: none"> <li>• Klammerausdrücke, <math>a^n b^n</math> im Vergleich zu <math>(ab)^n</math></li> </ul>
Zu entwickelnde Kompetenzen:	

Die Schülerinnen und Schüler

- analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens auf bestimmte Eingaben (A),
- analysieren und erläutern Grammatiken regulärer Sprachen (A),
- zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A),
- ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A),
- entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M),
- entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M),
- entwickeln zur akzeptierten Sprache eines Automaten die zugehörige Grammatik (M),
- entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M),
- modifizieren Grammatiken regulärer Sprachen (M),
- entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M),
- stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D),
- ermitteln die Sprache, die ein endlicher Automat akzeptiert (D).
- beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D).

**Unterrichtsvorhaben Q2-3**

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

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

*Zeitbedarf:* 12 Stunden

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

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

Sequenz 1: Von-Neumann-Architektur und die Ausführung maschinennaher Programme	Beispiele, Medien und Materialien
--	-----------------------------------

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

**Unterrichtsvorhaben Q1-4**

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

**2.4 Grundsätze der fachmethodischen und fachdidaktischen Arbeit**

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

**Überfachliche Grundsätze:**

1. Geeignete Problemstellungen zeichnen die Ziele des Unterrichts vor und bestimmen die Struktur der Lernprozesse.
2. Inhalt und Anforderungsniveau des Unterrichts entsprechen dem Leistungsvermögen der Schüler/innen.
3. Die Unterrichtsgestaltung ist auf die Ziele und Inhalte abgestimmt.
4. Medien und Arbeitsmittel sind schülernah gewählt.

5. Die Schüler/innen erreichen einen Lernzuwachs.
6. Der Unterricht fördert eine aktive Teilnahme der Schüler/innen.
7. Der Unterricht fördert die Zusammenarbeit zwischen den Schülern/innen und bietet ihnen Möglichkeiten zu eigenen Lösungen.
8. Der Unterricht berücksichtigt die individuellen Lernwege der einzelnen Schüler/innen.
9. Die Schüler/innen erhalten Gelegenheit zu selbstständiger Arbeit und werden dabei unterstützt.
10. Der Unterricht fördert strukturierte und funktionale Partner- bzw. Gruppenarbeit.
11. Der Unterricht fördert strukturierte und funktionale Arbeit im Plenum.
12. Die Lernumgebung ist vorbereitet; der Ordnungsrahmen wird eingehalten.
13. Die Lehr- und Lernzeit wird intensiv für Unterrichtszwecke genutzt.
14. Es herrscht ein positives pädagogisches Klima im Unterricht.

### **Fachliche Grundsätze:**

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

## **2.5 Grundsätze der Leistungsbewertung**

Auf der Grundlage von §13 - §16 der APO-GOST sowie Kapitel 3 des Kernlehrplans Informatik für die gymnasiale Oberstufe hat die Fachschaft Informatik 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.

### **2.5.1 Beurteilungsbereich Klausuren**

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

**Instrumente:**

- Einführungsphase: 1 Klausur je Halbjahr, Dauer: 2 Unterrichtsstunden
- Grundkurse Q 1: 2 Klausuren je Halbjahr, Dauer: 2 Unterrichtsstunden
- Grundkurse Q 2.1: 2 Klausuren, Dauer: 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 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.

**2.5.2 Beurteilungsbereich Sonstige Mitarbeit**

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

**Leistungsaspekte****Mündliche Leistungen**

- Beteiligung am Unterrichtsgespräch
- Zusammenfassungen zur Vor- und Nachbereitung des Unterrichts
- Präsentation von Arbeitsergebnissen
- Referate
- Mitarbeit in Partner-/Gruppenarbeitsphasen
- Praktische Leistungen am Computer
- Implementierung, Test und Anwendung von Informatiksystemen

**Sonstige schriftliche Leistungen**

- Bearbeitung von schriftlichen Aufgaben und Übungen 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.

### 2.5.3 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.

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

Es besteht die Möglichkeit im Rahmen der Facharbeit fächerübergreifende Themen zu bearbeiten, wobei im Rahmen der Facharbeit auch ein Produkt entwickelt werden sollte.

In der Einführungsphase oder in der Qualifikationsphase kann eine Exkursion zum Heinz Nixdorf MuseumsForum 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 ist zunächst bis 2017 für den ersten Durchgang durch die gymnasiale Oberstufe nach Erlass des Kernlehrplanes verbindlich. Erstmals nach Ende der Einführungsphase im Sommer 2015, 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.