

**Schulinterner Lehrplan  
zum Kernlehrplan für die gymnasiale Oberstufe**

**Informatik**

**(Stand: 06. Mai 2014)**

## **Inhalt**

|   | Seite     |
|---|-----------|
| <b>1 Die Fachgruppe Informatik des Gymnasium Broich in Mülheim</b>    | <b>3</b>  |
| <b>2 Entscheidungen zum Unterricht</b>                                | <b>5</b>  |
| 2.1 Unterrichtsvorhaben   | 5         |
| 2.1.1 Übersichtsraster Unterrichtsvorhaben                            | 6         |
| 2.1.2 Konkretisierte Unterrichtsvorhaben                              | 15        |
| 2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit       | 23        |
| 2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung        | 52        |
| <b>3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen</b> | <b>55</b> |
| <b>4 Qualitätssicherung und Evaluation</b>                            | <b>56</b> |

## 1 Die Fachgruppe Informatik des Gymnasiums Broich in Mülheim an der Ruhr

**Hinweis:** Um die Ausgangsbedingungen für die Erstellung des schulinternen Lehrplans festzuhalten, können beispielsweise folgende Aspekte berücksichtigt werden:

- Lage der Schule
- Funktionen und Aufgaben der Fachgruppe vor dem Hintergrund des Schulprogramms
- Beitrag der Fachgruppe zur Erreichung der Erziehungsziele ihrer Schule
- Beitrag zur Qualitätssicherung und –entwicklung innerhalb der Fachgruppe
- Zusammenarbeit mit anderen Fachgruppen (fächerübergreifende Unterrichtsvorhaben und Projekte)
- Ressourcen der Schule (personell, räumlich, sachlich), Größe der Lerngruppen, Unterrichtstaktung, Stundenverortung
- Name des/der Fachvorsitzenden und des Stellvertreters/der Stellvertreterin
- ggf. Arbeitsgruppen bzw. weitere Beauftragte

Beim Gymnasium Broich handelt es sich um eine vierzügige Schule im Stadtteil Broich in Mülheim an der Ruhr mit zurzeit ca. 900 Schülerinnen und Schüler sowie ca. 60 Lehrerinnen und Lehrern. Das Einzugsgebiet umfasst den Teil linksseits der Ruhr sowie vereinzelt auch Schülerinnen und Schüler aus anderen Teilen Mülheims und Umgebung.

Im Bereich der Sekundarstufe II kooperiert das Gymnasium Broich mit seinen Nachbarschulen und bietet mit ihnen zahlreiche gemeinsame Kurse an.

Das Fach Informatik wird am Gymnasium Broich ab der Jahrgangsstufe 5 mit dem Unterrichtsfach ITG (Informationstechnische Grundbildung) unterrichtet. Hierbei erlernen die Schülerinnen und Schüler in altersgerechter Weise die Grundzüge der Netzwerktechnik am Beispiel unseres Schulnetzwerkes und wie dieses von ihnen sinnvoll genutzt werden kann. Außerdem lernen sie den Umgang mit Office-Programmen sowie die Möglichkeiten und Gefahren des Internets kennen.

Weitergeführt wird das Fach Informatik durch den im Wahlpflichtbereich II (WP II) zweistündigen Unterricht des Unterrichtsfaches Mathematik/Informatik. Dieser Wahlpflichtbereich wird von etwa der Hälfte der Schülerinnen und Schüler der Jahrgangsstufen 8/9 besucht. In der zweijährigen Laufzeit dieser Kurse werden in altersstufengerechter Weise unter anderem sowohl seitenbeschreibende, prozedurale, als auch objektori-

enterte Programmieransätze aufgezeigt. Ebenso wird auch auf die technische Informatik sowie die damit verbundene Robotik eingegangen. Die Schülerinnen und Schüler haben die Möglichkeit sich durch die Arbeit mit Lego Robotern und dem didaktisch durchdachten Roberta-Konzept in diesen Themenbereich einzuarbeiten.

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

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

Der Unterricht der Sekundarstufe II wird mit Hilfe der Programmiersprache Java durchgeführt. Als Editor dient ihnen dabei die Programmierumgebung blueJ die extra für den didaktischen Einsatz entwickelt wurde um den Schülerinnen und Schülern den Einstieg zu erleichtern.

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 Gymnasium Broich aus zwei Lehrkräften, wobei eine von ihnen ausschließlich die Lehrbefähigung für die Sekundarstufe I besitzt. Ihnen stehen vier feste Computerräume mit drei mal 15 und einmal 14 Computerarbeitsplätzen, ein mobiler Computerraum mit 15 Computerarbeitsplätzen und ein Selbstlernzentrum mit 7 Plätzen zur Verfügung. Alle Arbeitsplätze sind an das schulinterne Rechnernetz angeschlossen, so dass Schülerinnen und Schüler über einen individuell gestaltbaren Zugang zum zentralen Server der Schule an allen Arbeitsplätzen Zugriff auf ihre eigenen Daten haben, die Computer zur Recherche im Internet oder zur Bearbeitung schulischer Aufgaben nutzen können.

Der Unterricht erfolgt im 90-Minuten-Takt. Mit jeweils zwei Doppelstunden in der einen Woche und einer Doppelstunde in der anderen Woche.

## 2 Entscheidungen zum Unterricht

### 2.1 Unterrichtsvorhaben

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

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

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

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

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

## 2.1.1 Übersichtsraster Unterrichtsvorhaben

### I) Einführungsphase

| Einführungsphase  |   |
|---|---|
| <p><u>Unterrichtsvorhaben E-I</u></p> <p><b>Thema:</b><br/><i>Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"><li>• Argumentieren</li><li>• Darstellen und Interpretieren</li><li>• Kommunizieren und Kooperieren</li></ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"><li>• Informatiksysteme</li><li>• Informatik, Mensch und Gesellschaft</li></ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"><li>• Einzelrechner</li><li>• Dateisystem</li><li>• Internet</li><li>• Einsatz von Informatiksystemen</li></ul> <p><b>Zeitbedarf:</b> 3 Stunden</p> | <p><u>Unterrichtsvorhaben E-II</u></p> <p><b>Thema:</b><br/><i>Grundlagen der objektorientierten Analyse, Modellierung und Implementierung anhand der Erweiterung vorgegebener Klassen mit grafischer Darstellung</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"><li>• Modellieren</li><li>• Implementieren</li><li>• Darstellen und Interpretieren</li><li>• Kommunizieren und Kooperieren</li></ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"><li>• Daten und ihre Strukturierung</li><li>• Diagramme</li><li>• Formale Sprachen</li></ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"><li>• Objekte und Klassen</li><li>• Syntax und Semantik einer Programmiersprache</li></ul> <p><b>Zeitbedarf:</b> 8 Stunden</p> |

## Einführungsphase

### Unterrichtsvorhaben E-III

**Thema:**

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

**Zentrale Kompetenzen:**

- Argumentieren
- Modellieren
- Implementieren
- Kommunizieren und Kooperieren

**Inhaltsfelder:**

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

**Inhaltliche Schwerpunkte:**

- Objekte und Klassen
- Syntax und Semantik einer Programmiersprache
- Analyse, Entwurf und Implementierung einfacher Algorithmen

**Zeitbedarf:** 36 Stunden

### Unterrichtsvorhaben E-IV

**Thema:**

*Such- und Sortieralgorithmen anhand kontextbezogener Beispiele*

**Zentrale Kompetenzen:**

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

**Inhaltsfelder:**

- Algorithmen
- Daten und ihre Strukturierung

**Inhaltliche Schwerpunkte:**

- Algorithmen zum Suchen und Sortieren
- Analyse, Entwurf und Implementierung einfacher Algorithmen

**Zeitbedarf:** 12 Stunden

## Einführungsphase

### Unterrichtsvorhaben E-V

**Thema:**

*Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes*

**Zentrale Kompetenzen:**

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

**Inhaltsfelder:**

- Informatik, Mensch und Gesellschaft
- Informatiksysteme

**Inhaltliche Schwerpunkte:**

- Wirkungen der Automatisierung
- Geschichte der automatischen Datenverarbeitung
- Digitalisierung

**Zeitbedarf:** 12 Stunden

**Summe Einführungsphase: 74**



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

| Qualifikationsphase 1  |  |
|--|--|
| <p><u>Unterrichtsvorhaben Q1-I</u></p> <p><b>Thema:</b><br/><i>Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>• Argumentieren</li> <li>• Modellieren</li> <li>• Implementieren</li> <li>• Darstellen und Interpretieren</li> <li>• Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>• Daten und ihre Strukturierung</li> <li>• Algorithmen</li> <li>• Formale Sprachen und Automaten</li> <li>• Informatiksysteme</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>• Objekte und Klassen</li> <li>• Analyse, Entwurf und Implementierung von Algorithmen</li> <li>• Syntax und Semantik einer Programmiersprache</li> <li>• Nutzung von Informatiksystemen</li> </ul> <p><b>Zeitbedarf:</b> 8 Stunden</p> | <p><u>Unterrichtsvorhaben Q1-II</u></p> <p><b>Thema:</b><br/><i>Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>• Argumentieren</li> <li>• Modellieren</li> <li>• Implementieren</li> <li>• Darstellen und Interpretieren</li> <li>• Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>• Daten und ihre Strukturierung</li> <li>• Algorithmen</li> <li>• Formale Sprachen und Automaten</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>• Objekte und Klassen</li> <li>• Analyse, Entwurf und Implementierung von Algorithmen</li> <li>• Algorithmen in ausgewählten informatischen Kontexten</li> <li>• Syntax und Semantik einer Programmiersprache</li> </ul> <p><b>Zeitbedarf:</b> 14 Stunden</p> |

## Qualifikationsphase 1

### Unterrichtsvorhaben Q1-III

**Thema:**

*Suchen und Sortieren auf linearen Datenstrukturen*

**Zentrale Kompetenzen:**

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

**Inhaltsfelder:**

- Algorithmen
- Formale Sprachen und Automaten

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 6 Stunden

### Unterrichtsvorhaben Q1-IV

**Thema:**

*Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen wie Binärbäumen*

**Zentrale Kompetenzen:**

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

**Inhaltsfelder:**

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

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 12 Stunden

## Qualifikationsphase 1

### Unterrichtsvorhaben Q1-V

**Thema:**

*Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen wie Graphen*

**Zentrale Kompetenzen:**

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

**Inhaltsfelder:**

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

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 14 Stunden

### Unterrichtsvorhaben Q1-VI

**Thema:**

*Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten*

**Zentrale Kompetenzen:**

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

**Inhaltsfelder:**

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

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 20 Stunden

## Qualifikationsphase 1

### Unterrichtsvorhaben Q1-VII

**Thema:**

*Sicherheit und Datenschutz in Netzstrukturen*

**Zentrale Kompetenzen:**

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

**Inhaltsfelder:**

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 6 Stunden

**Summe Qualifikationsphase 1: 74 Stunden**

## Qualifikationsphase 2

### Unterrichtsvorhaben Q2-I

**Thema:**

*Endliche Automaten und formale Sprachen*

**Zentrale Kompetenzen:**

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

**Inhaltsfelder:**

- Endliche Automaten und formale Sprachen

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 24 Stunden

### Unterrichtsvorhaben Q2-II

**Thema:**

*Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit*

**Zentrale Kompetenzen:**

- Argumentieren
- Kommunizieren und Kooperieren

**Inhaltsfelder:**

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

**Inhaltliche Schwerpunkte:**

- Einzelrechner und Rechnernetzwerke
- Grenzen der Automatisierung

**Zeitbedarf:** 12 Stunden

## Qualifikationsphase 2

### Unterrichtsvorhaben Q2-III

**Thema:**

*Verschlüsselungsverfahren bei der Kommunikation sowie der zugehörige Datenschutz*

**Zentrale Kompetenzen:**

- Argumentieren
- Kommunizieren und Kooperieren

**Inhaltsfelder:**

- Kryptographie
- Informatiksysteme
- Informatik, Mensch und Gesellschaft

**Inhaltliche Schwerpunkte:**

- Geschichte der Verschlüsselung
- Verschlüsselungsverfahren und der Schlüsselaustausch
- Rechnernetzwerke
- Grenzen der Kryptographie

**Zeitbedarf: 20 Stunden**

**Summe Qualifikationsphase 2: 56 Stunden**

---

## 2.1.2 Konkretisierte Unterrichtsvorhaben

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

In der Sekundarstufe II arbeiten die Schülerinnen und Schüler mit der Entwicklungsumgebung blueJ und den durch den Dudenverlag vorgegebenen Klassenbibliotheken.

- Download der Software: <http://www.bluej.org/>
- Download der Klassenbibliotheken und vordefinierten Klassen und Projekte (Anmeldung erforderlich): <http://duden-paetec.de/informatik/>

In der Sekundarstufe II werden die Unterrichtsvorhaben unter Berücksichtigung der Vorgaben für das Zentralabitur Informatik in NRW konkretisiert. Diese sind zu beziehen unter der Adresse

<http://www.standardsicherung.schulministerium.nrw.de/abitur-gost/fach.php?fach=15> (abgerufen: 06. Mai 2014)

---

## I) Einführungsphase

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

Die Schülerinnen und Schüler

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

### Unterrichtsvorhaben EF-I

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

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

**Vorhabenbezogene Konkretisierung:**

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

**Zeitbedarf:** 3 Stunden



## Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen  | Zu entwickelnde Kompetenzen  | Beispiele, Medien, Materialien  |
|---|--|---|
| <p><b>1. Information, deren Kodierung und Speicherung</b></p> <ul style="list-style-type: none"> <li>(a) Informatik als Wissenschaft der Verarbeitung von Informationen</li> <li>(b) Darstellung von Informationen in Schrift, Bild und Ton</li> <li>(c) Speichern von Daten mit informatischen Systemen am Beispiel der Schulrechner</li> <li>(d) Vereinbarung von Richtlinien zur Datenspeicherung auf den Schulrechnern und dem nur für den Informatikunterricht genutzten USB-Stick (Ordnerstruktur, Dateibezeichner usw.)</li> </ul> | <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• beschreiben und erläutern die Arbeitsweise singulärer Rechner zum Beispiel durch die „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> | <p><i>Beispiel:</i> Textkodierung<br/>Kodierung und Dekodierung von Texten mit unbekanntem Zeichensätzen (z.B. Wingdings)</p> <p><i>Beispiel:</i> Bildkodierung<br/>Kodierung von Bildinformationen in Raster- und Vektorgrafiken</p> <p><i>Beispiel:</i> Kettengeschichte<br/>Ergänzen von Texten die über das schulinterne Netz ausgetauscht werden</p>   |
| <p><b>2. Informations- und Datenübermittlung in Netzen</b></p> <ul style="list-style-type: none"> <li>(a) „Sender-Empfänger-Modell“ und seine Bedeutung für die Eindeutigkeit von Kommunikation</li> <li>(b) Informatische Kommunikation in Rechnernetzen am Beispiel des Schulnetzwerks (z.B. Benutzeranmeldung, Netzwerkordner, Zugriffsrechte, Client-Server)</li> <li>(c) Richtlinien zum verantwortungsvollen Umgang mit dem Internet</li> </ul>   |  | <p><i>Beispiel:</i> Rollenspiel zur Paketvermittlung im Internet<br/>Schülerinnen und Schüler übernehmen die Rollen von Clients und Routern. Sie schicken spielerisch Informationen auf Karten von einem Schüler-Client zum anderen. Jede Schülerin und jeder Schüler hat eine Adresse, jeder Router darüber hinaus eine Routingtabelle. Mit Hilfe der Tabelle und einem Würfel wird entschieden, wie ein Paket weiter vermittelt wird.</p> |

|   |  |   |
|---|--|---|
| <p><b>3. Aufbau informatischer Systeme</b></p> <p>(a) Identifikation typischer Komponenten informatischer Systeme und anschließende Beschränkung auf das Wesentliche, Herleitung der „Von-Neumann-Architektur“</p> <p>(b) Identifikation des EVA-Prinzips (Eingabe-Verarbeitung-Ausgabe) als Prinzip der Verarbeitung von Daten und Grundlage der „Von-Neumann-Architektur“</p> |  | <p><i>Material:</i> Demonstrationshardware</p> <p>Durch Demontage eines Demonstrationsrechners entdecken Schülerinnen und Schüler die verschiedenen Hardwarekomponenten eines Informatiksystems. Als Demonstrationsrechner bietet sich ein ausrangierter Schulrechner an.</p> |
|---|--|---|

## Unterrichtsvorhaben EF-II

**Thema:** Grundlagen der objektorientierten Analyse, Modellierung und Implementierung anhand der Erweiterung vorgegebener Klassen mit grafischer Darstellung

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

### Vorhabenbezogene Konkretisierung:

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

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

Im Anschluss werden die von der Bibliothek vorgegebenen Klassen von Schülerinnen und Schülern in Teilen analysiert und entsprechende Objekte anhand einfacher Problemstellungen erprobt. Dazu muss der grundlegende Aufbau einer Java-Klasse thematisiert und zwischen Deklaration, Initialisierung und Methodenaufrufen unterschieden werden.

Da bei der Umsetzung dieser ersten Projekte konsequent auf die Verwendung von Kontrollstrukturen verzichtet wird und der Quellcode aus einer rein linearen Sequenz besteht, ist auf diese Weise eine Fokussierung auf die Grundlagen der Objektorientierung möglich, ohne dass

algorithmische Probleme ablenken. Natürlich kann die Arbeit an diesen Projekten unmittelbar zum nächsten Unterrichtsvorhaben führen. Dort stehen unter anderem Kontrollstrukturen im Mittelpunkt.

**Zeitbedarf:** 8 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

| Unterrichtssequenzen   | Zu entwickelnde Kompetenzen   | Beispiele, Medien, Materialien   |
|--|---|--|
| <p><b>1. Identifikation von Objekten</b></p> <p>(a) Am Beispiel eines lebensweltnahen Beispiels werden Objekte im Sinne der Objektorientierten Modellierung eingeführt.</p> <p>(b) Objekte werden mit Objektkarten visualisiert und mit sinnvollen Attributen und „Fähigkeiten“, d.h. Methoden versehen.</p> <p>(c) Manche Objekte sind prinzipiell typgleich und werden so zu einer Objektsorte bzw. Objektklasse zusammengefasst.</p> <p>(d) Vertiefung: Modellierung weiterer Beispiele ähnlichen Musters</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>• modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M),</li> <li>• stellen die Kommunikation zwischen Objekten grafisch dar (M),</li> <li>• implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik der Programmiersprache Java (I),</li> <li>• stellen den Zustand eines Objekts dar (D).</li> </ul> | <p><i>Beispiel:</i> Autohaus<br/>Schülerinnen und Schüler betrachten das Angebot der Autos als Menge gleichartiger Objekte, die in einer Klasse mit Attributen und Methoden zusammengefasst werden können.</p> <p>Beispiel: Schatzsuche<br/>Schülerinnen und Schüler stilisieren eine Schatzkarte indem die Schatzsucher ihren Zustand (aktuelle Position) in einen Zustandsdiagramm darstellen.</p> <p><i>Materialien:</i><br/>Vorlagen Kapitel 1 und 2</p> |
| <p><b>2. Analyse von Klassen didaktischer Lernumgebungen</b><br/>Objektorientierte Programmierung als modularisiertes Vorgehen (Entwicklung von Problemlösungen auf Grundlage vorhandener Klassen)</p>   |   |  |

---

## Unterrichtsvorhaben EF-III

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

**Leitfragen:** *Wie lassen sich Animationen und Simulationen optischer Gegenstandsbereiche unter Berücksichtigung von Methodenaufrufen realisieren?*

### **Vorhabenbezogene Konkretisierung:**

Der Schwerpunkt dieses Unterrichtsvorhabens liegt auf der Erweiterung mehrerer Projekte, die durch Aktionen des Benutzers gesteuerte Animationen aufweisen. Zunächst wird ein Projekt bearbeitet, bei dem ein Roboterarm eines Fabrikroboters am Fließband mehrere Aufgaben erfüllen muss. Der Roboterarm ist stilisiert durch einfache grafische Objekte dargestellt und kann auf vordefinierte Methoden agieren. Für die Umsetzung dieses Projekts werden Kontrollstrukturen in Form von Schleifen und Verzweigungen benötigt und eingeführt.

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

Das Unterrichtsvorhaben schließt mit einem Projekt, das die Abläufe eines Geldautomaten simuliert, bei der die Schülerinnen und Schüler alle ihnen bekannten Operationen auf einem Array, sowie Schleifen und Verzweigungen anwenden müssen.

Komplexere Assoziationsbeziehungen zwischen Klassen werden in diesem Unterrichtsvorhaben zunächst nicht behandelt. Sie stellen den Schwerpunkt des folgenden Vorhabens dar.

**Zeitbedarf:** 16 Stunden

## Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen  | Zu entwickelnde Kompetenzen   | Beispiele, Medien, Materialien  |
|---|---|---|
| <p><b>1. Bewegungsanimationen am Beispiel einfacher vorgegebener grafischer Objekte</b></p> <p>(a) Kontinuierliche Suche nach einem Objekt, das eine bestimmte Bedingung erfüllt, mit Hilfe einer Schleife (While-Schleife)</p> <p>(b) Festgelegte Anzahl an Wiederholungen eines Algorithmus (For-Schleife)</p> <p>(c) Mehrstufige Animationen mit mehreren sequenziellen Schleifen</p> <p>(d) Berechnung von Mittelwert, Minimum und Maximum mit Hilfsvariablen</p> | <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• analysieren und erläutern einfache Algorithmen und Programme (A),</li> <li>• entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M),</li> <li>• ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li> <li>• modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M),</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M),</li> <li>• ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),</li> <li>• modifizieren einfache Algorithmen und Programme (I),</li> <li>• implementieren Klassen in der Programmiersprache Java auch unter Nutzung dokumentierter Klassenbibliotheken (I),</li> <li>• implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I),</li> <li>• implementieren einfache Algorithmen unter Beachtung der Syntax und Seman-</li> </ul> | <p><i>Beispiel:</i> Fabrikroboter<br/>Die Schülerinnen und Schüler erweitern ein Programm so, dass ein stilisierter Roboterarm vorgegebene Aufgaben erfüllen kann.</p> <p><i>Materialien:</i><br/>Vordefinierte Klassen Kapitel 3</p> |
| <p><b>2. Erstellen und Verwalten größerer Mengen von Objekten</b></p> <p>(a) Erzeugung von Objekten mit Hilfe von Zählschleifen (FOR-Schleife)</p> <p>(b) Verwaltung von Objekten in eindimensionalen Feldern (Arrays)</p> <p>(c) Ansprechen von Objekten, die in eindimensionalen Feldern (Arrays) verwaltet werden</p> <p>(d) Vertiefung: Verschiedene Feldbeispiele</p>  | <ul style="list-style-type: none"> <li>• ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),</li> <li>• modifizieren einfache Algorithmen und Programme (I),</li> <li>• implementieren Klassen in der Programmiersprache Java auch unter Nutzung dokumentierter Klassenbibliotheken (I),</li> <li>• implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I),</li> <li>• implementieren einfache Algorithmen unter Beachtung der Syntax und Seman-</li> </ul>  |   |

|  |   |  |
|--|---|--|
|  | tik einer Programmiersprache (I), <ul style="list-style-type: none"> <li>• testen Programme schrittweise anhand von Beispielen (I),</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I).</li> </ul> |  |
|--|---|--|

## Unterrichtsvorhaben EF-IV

**Thema:** Such- und Sortieralgorithmen anhand kontextbezogener Beispiele

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

### **Vorhabenbezogene Konkretisierung:**

Dieses Unterrichtsvorhaben beschäftigt sich mit der Erarbeitung von Such- und Sortieralgorithmen. Der Schwerpunkt des Vorhabens liegt dabei auf den Algorithmen selbst und nicht auf deren Implementierung in einer Programmiersprache. Lediglich das Bubblesort-Verfahren soll als ein Beispiel implementiert werden.

Zunächst erarbeiten die Schülerinnen und Schüler mögliche Einsatzszenarien für Such- und Sortieralgorithmen, um sich der Bedeutung einer effizienten Lösung dieser Probleme bewusst zu werden. Anschließend werden Strategien zur Sortierung mit Hilfe eines explorativen Spiels von den Schülerinnen und Schülern selbst erarbeitet und hinsichtlich der Anzahl notwendiger Vergleiche auf ihre Effizienz untersucht.

Daran anschließend werden die erarbeiteten Strategien systematisiert und im Pseudocode notiert. Die Schülerinnen und Schüler sollen auf diese Weise das *Sortieren durch Vertauschen*, das *Sortieren durch Auswählen* und mindestens einen weiteren Sortieralgorithmus, kennen lernen.

**Zeitbedarf:** 12 Stunden

## Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen  | Zu entwickelnde Kompetenzen  | Beispiele, Medien, Materialien  |
|---|--|---|
| <p><b>1. Explorative Erarbeitung eines Sortierverfahrens</b></p> <p>(a) Sortierprobleme im Kontext informatischer Systeme und im Alltag (z.B. Dateisortierung, Tabellenkalkulation, Telefonbuch, Bundesligatabelle, usw.)</p> <p>(b) Vergleich zweier Elemente als Grundlage eines Sortieralgorithmus</p> <p>(c) Erarbeitung eines Sortieralgorithmus durch die Schülerinnen und Schüler</p>  | <p>Die Schülerinnen und Schüler</p> <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> | <p><i>Beispiel:</i> Sortieren mit Waage<br/>Die Schülerinnen und Schüler bekommen die Aufgabe, kleine, optisch identische Kunststoffbehälter aufsteigend nach ihrem Gewicht zu sortieren. Dazu steht ihnen eine Balkenwaage zur Verfügung, mit deren Hilfe sie das Gewicht zweier Behälter vergleichen können.</p>  |
| <p><b>2. Systematisierung von Algorithmen und Effizienzbetrachtungen</b></p> <p>(a) Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode (auf jeden Fall: Sortieren durch Vertauschen, Sortieren durch Auswählen)</p> <p>(b) Anwendung von Sortieralgorithmen auf verschiedene Beispiele</p> <p>(c) Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche</p> <p>(d) Variante des Sortierens durch Auswählen (Nutzung eines einzigen oder zweier Felder bzw. lediglich eines einzigen zusätzlichen Ablageplatzes oder mehrerer neuer Ablageplätze)</p> <p>(e) Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des Speicherplatzbedarfs</p> |  | <p><i>Beispiele:</i> Sortieren durch Auswählen, Sortieren durch Vertauschen, Quicksort<br/>Quicksort ist als Beispiel für einen Algorithmus nach dem Prinzip <i>Teile und Herrsche</i> gut zu behandeln. Kenntnisse in rekursiver Programmierung sind nicht erforderlich, da eine Implementierung nicht angestrebt wird.</p> <p><i>Beispiel:</i><br/>Die Schülerinnen und Schüler werden von einem Mitschüler so sortiert, dass sie der Größe nach in einer Reihe stehen. Strategie besprechen und Möglichkeiten der Implementation erarbeiten. Anschließend implementieren die Schülerinnen und Schüler ihre Idee.</p> |

|   |  |  |
|---|--|--|
| (f) Analyse des weiteren Sortieralgorithmus (sofern nicht in Sequenz 1 und 2 bereits geschehen)<br>(g) Implementierung des Bubblesort-Algorithmus unter Verwendung eines Arrays |  |  |
|---|--|--|

## Unterrichtsvorhaben EF-V

**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?*

### **Vorhabenbezogene Konkretisierung:**

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 Form von Plakatpräsentationen vorgestellt. 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.

**Zeitbedarf:** 12 Stunden



## Sequenzierung des Unterrichtsvorhabens:

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

---

## II) Qualifikationsphase

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

Die Schülerinnen und Schüler

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

### Unterrichtsvorhaben Q1-I:

**Thema:** Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung

**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?*

#### **Vorhabenbezogenen Konkretisierung:**

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 verschie-

denen Objekten wird verdeutlicht, indem die Kommunikation zwischen zwei ausgewählten Objekten grafisch dargestellt wird. In diesem Zusammenhang wird das Nachrichtenkonzept der objektorientierten Programmierung wiederholt.

**Zeitbedarf:** 8 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

| Unterrichtssequenzen   | Zu entwickelnde Kompetenzen  | Beispiele, Medien, Materialien   |
|--|--|--|
| <p><b>1. Wiederholung und Erweiterung der objektorientierten Modellierung und Programmierung durch Analyse und Erweiterung eines kontextbezogenen Beispiels</b></p> <p>(a) Analyse der Problemstellung</p> <p>(b) Analyse der Modellierung (Implementationsdiagramm)</p> <p>(c) Erweiterung der Modellierung im Implementationsdiagramm (Vererbung, abstrakte Klasse)</p> <p>(d) Kommunikation zwischen mindestens zwei Objekten (grafische Darstellung)</p> <p>(e) Dokumentation von Klassen</p> <p>(f) Implementierung der Anwendung oder von Teilen der Anwendung</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> </ul> | <p><i>Beispiel:</i> Taxiwarteschlange</p> <p>An einem Taxistand stehen mehrere Taxen bei denen die An- und Abfahrt simuliert werden soll. Die Schülerinnen und Schüler implementieren die benötigten Klassen und setzen ihre Beziehungen um.</p> |

|  |   |  |
|--|---|--|
|  | <ul style="list-style-type: none"> <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 Q1-II:

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

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

### Vorhabenbezogene Konkretisierung:

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.

**Zeitbedarf:** 20 Stunden

## Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen  | Zu entwickelnde Kompetenzen   | Beispiele, Medien, Materialien   |
|---|---|--|
| <p><b>1. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse <code>Queue</code></b></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse <code>Queue</code></p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse <code>Queue</code></p>          | <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• 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> </ul> | <p><i>Beispiel:</i> Patientenwarteschlange (jeder kennt seinen Nachfolger bzw. alternativ: seinen Vorgänger)</p> <p>Sobald ein Patient in einer Arztpraxis eintrifft, werden sein Name und seine Krankenkasse erfasst. Die Verwaltung der Patientenwarteschlange geschieht über eine Klasse, die hier als Wartezimmer bezeichnet wird. Wesentliche Operationen sind das „Hinzufügen“ eines Patienten und das „Entfernen“ eines Patienten, wenn er zur Behandlung gerufen wird.</p> <p>Anschließend wird der Funktionsumfang der Anwendung erweitert: Patienten können sich zusätzlich in die Warteschlange zum Blutdruckmessen einreihen. Objekte werden von zwei Schlangen verwaltet.</p> |
| <p><b>2. Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse <code>Stack</code></b></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse <code>Stack</code></p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse <code>Stack</code></p> <p>(d)</p> | <p>(I),</p> <ul style="list-style-type: none"> <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> </ul>   | <p><i>Beispiel:</i> Heftstapel</p> <p>In einem Heftstapel soll das Heft einer Schülerin gefunden werden.</p> <p>oder</p> <p><i>Beispiel:</i> Kisten stapeln</p> <p>In einem Stapel nummerierter Kisten soll eine bestimmte Kiste gefunden und an einen Kunden geliefert werden. Dazu müssen Kisten auf verschiedene Stapel gestapelt und wieder zurückgestellt werden.</p>   |

|   |  |  |
|---|--|--|
| <p><b>3. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse <code>List</code></b></p> <p>(a) Erarbeitung der Vorteile der Klasse <code>List</code> im Gegensatz zu den bereits bekannten linearen Strukturen</p> <p>(b) Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse <code>List</code>.</p> | <ul style="list-style-type: none"> <li>stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D).</li> </ul> |  |
| <p><b>4. Vertiefung - Anwendungen von Listen, Stapeln oder Schlangen in mindestens einem weiteren Kontext</b></p>   |  | <p><i>Beispiel:</i> Skispringen<br/> Ein Skispringen hat folgenden Ablauf: Nach dem Sprung erhält der Springer eine Punktzahl und wird nach dieser Punktzahl in eine Rangliste eingeordnet. Die besten 30 Springer qualifizieren sich für den zweiten Durchgang. Sie starten in umgekehrter Reihenfolge gegenüber der Platzierung auf der Rangliste. Nach dem Sprung erhält der Springer wiederum eine Punktzahl und wird nach der Gesamtpunktzahl aus beiden Durchgängen in die endgültige Rangliste eingeordnet.</p> |

## Unterrichtsvorhaben Q1-III:

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

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

### Vorhabenbezogene Konkretisierung:

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 Implementationen 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 implementieren die Schülerinnen und Schüler eine lineare verkettete Liste auf der Grundlage der Kompositums.

**Zeitbedarf:** 6 Stunden

### Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen  | Zu entwickelnde Kompetenzen   | Beispiele, Medien, Materialien   |
|---|---|--|
| <b>1. Suchen von Daten in Listen und Arrays</b><br>(a) Lineare Suche in Listen und in Arrays<br>(b) Binäre Suche in Arrays als Beispiel für rekursives Problemlösen | Die Schülerinnen und Schüler <ul style="list-style-type: none"><li>• analysieren und erläutern Algorithmen und Programme (A),</li><li>• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li><li>• beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A),</li><li>• entwickeln iterative und rekursive Algo-</li></ul> | <i>Beispiel:</i> Karteiverwaltung<br>Für ein Adressverwaltungsprogramm soll eine Methode zum Suchen einer Adresse geschrieben werden.<br><br>oder<br><br><i>Beispiel:</i> Obstkette<br>Mehrere Körbe mit Obst werden zu einer Kette verbunden. Der zugriff auf das Obst erfolgt immer über die Aufhänge Vorrichtung. |

**2. Sortieren in Listen und Arrays - Entwicklung und Implementierung von iterativen und rekursiven Sortierverfahren**

- (a) Entwicklung und Implementierung eines einfachen Sortierverfahrens für eine lineare Liste
- (b) Implementierung des Löschens und hinzufügen von Objekten zur linearen Liste

- rithmen 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-IV:

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

**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?*

### Vorhabenbezogene Konkretisierung:

In einem Anwendungskontext werden zunächst die Familienstammbäume mit Mutter und Vater grafisch dargestellt. Hierzu werden Verfahren entwickelt und implementiert bzw. analysiert und erläutert.

Anschließend werden Verfahren entwickelt und implementiert mit denen Daten in einem geordneten, Suchbaum gespeichert, gelöscht und ausgelesen werden können. Hierbei soll auch ein rekursives Suchverfahren entwickelt werden. Die drei Traversierungsverfahren Preorder, Postorder und Inorder werden analysiert und angewendet.

Abschließend wird als eine Spezialform, der AVL-Baum, behandelt. Hierbei werden seine speziellen Eigenschaften untersucht und der Aufbau eines ausgeglichenen AVL-Baums erarbeitet. So erhalten die Schülerinnen und Schüler einen Einblick in die Effizienz verschiedener Datenstrukturen.

**Zeitbedarf:** 12 Stunden

### Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen  | Zu entwickelnde Kompetenzen  | Beispiele, Medien, Materialien   |
|---|--|--|
| <b>1. Analyse von Baumstrukturen in verschiedenen Kontexten</b><br>(a) Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit)<br>(b) Aufbau und Darstellung von binären Bäumen anhand von Baumstrukturen in verschiedenen Kontexten | Die Schülerinnen und Schüler <ul style="list-style-type: none"><li>• analysieren und erläutern Algorithmen und Programme (A),</li><li>• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li><li>• modifizieren Algorithmen und Programme (I),</li></ul> | <i>Beispiel:</i> Stammbaum<br>Um sich einen Überblick über seine Ahnen zu verschaffen erstellen die Schülerinnen und Schüler einen Stammbaum ihrer Familie.<br><br>oder<br><i>Beispiel:</i> Suchbäume (zur sortierten Speicherung von Daten) |

|  |  |  |
|--|--|--|
| <p><b>2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse <code>BinaryTree</code></b></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms</p> <p>(c) Erarbeitung der Klasse <code>BinaryTree</code> und beispielhafte Anwendung der Operationen</p> <p>(d) Implementierung der Anwendung oder von Teilen der Anwendung</p> <p>(e) Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf</p> | <ul style="list-style-type: none"> <li>• implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),</li> <li>• implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (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 iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).</li> </ul> | <p>Alle Inhalte, die nach einer Ordnung vor dem Inhalt im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem Inhalt im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.)</p> <p><i>oder</i></p> <p><i>Beispiel: Entscheidungsbäume</i><br/>Um eine Entscheidung zu treffen, werden mehrere Fragen mit ja oder nein beantwortet. Die Fragen, die möglich sind, wenn die Antwort auf eine Frage mit „ja“ beantwortet wird, befinden sich im linken Teilbaum, die Fragen, die möglich sind, wenn die Antwort „nein“ lautet, stehen im rechten Teilbaum.</p> |
| <p><b>3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse <code>BinarySearchTree</code></b></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramm, grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften</p> <p>(c) Erarbeitung der Klasse <code>BinarySearchTree</code> und Einführung des Interface <code>Item</code> zur</p>  |  |  |

---

|  |  |  |
|--|--|--|
| Realisierung einer geeigneten Ordnungsrelation<br>(d) Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums |  |  |
|--|--|--|

## Unterrichtsvorhaben Q1-V:

**Thema:** Modellierung und Implementierung mit dynamischen, nichtlinearen Datenstrukturen wie Graphen

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

### Vorhabenbezogene Konkretisierung:

Ein Auszug des Münchener U- und S-Bahnnetzes dient als Anwendungskontext zur Erarbeitung einfacher Graphen.

Hierzu werden Eigenschaften entwickelt und implementiert bzw. analysiert und erläutert, die aufzeigen, dass die bisher bekannten Datenstrukturen zur Analyse eines solchen Netzes nicht sinnvoll sind.

Anschließend werden Implementationsmöglichkeiten gesammelt, um auf die Adjazenzmatrix zu kommen, die eine Möglichkeit der Implementation darstellt. Diese wird in Java umgesetzt um darauf sowohl die Tiefen-, als auch Breitensuche anzuwenden, verschiedene Algorithmen zur Bestimmung des Minimalen Spannbaum sowie die Kürzesten Wege zu erarbeiten.

Der Schwerpunkt dieses Vorhabens liegt dabei auf den Algorithmen selbst und nicht auf deren Implementierung in einer Programmiersprache, deshalb wird dies auch nur am Rande behandelt.

**Zeitbedarf:** 14 Stunden

### Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen  | Zu entwickelnde Kompetenzen   | Beispiele, Medien, Materialien   |
|---|---|--|
| <b>1. Implementierung eines Graphen</b><br>Vorgegebener Quelltext wird analysiert und erweitert   | Die Schülerinnen und Schüler <ul style="list-style-type: none"><li>• analysieren und erläutern Algorithmen und Programme (A),</li><li>• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li><li>• modifizieren Algorithmen und Programme (I),</li><li>• implementieren iterative und rekursive Algorithmen auch unter Verwendung von</li></ul> | <i>Beispiel:</i> Labyrinth<br>Der Weg durch ein Labyrinth wird gesucht. Die Schülerinnen und Schüler erarbeiten sich möglich Wege durch das Labyrinth. |
| <b>2. Suchen von Daten in Graphen</b><br>(a) Tiefensuche<br>(b) Breitensuche<br>(c) Kürzeste Wege |   | <i>Beispiel:</i> Höhlenforscher<br>Eine Gruppe erforscht eine Höhe mit Hilfe der Tiefensuche.  |
| <b>3. Aufbau von Graphen</b>  |   |  |

|   |   |  |
|---|---|--|
| <p>(a) Minimaler Spannbaum durch Prim und Kruskal</p> <p>(b) Implementierung einer der gelernten Algorithmen</p> <p>(c)</p> | <p>dynamischen Datenstrukturen (I),</p> <ul style="list-style-type: none"><li>• implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (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 iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).</li></ul> |  |
|---|---|--|

---

## Unterrichtsvorhaben Q1-VI:

**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?*

### **Vorhabenbezogene Konkretisierung:**

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.

**Zeitbedarf:** 20 Stunden

## Sequenzierung des Unterrichtsvorhabens

| Unterrichtssequenzen  | Zu entwickelnde Kompetenzen   | Beispiele, Medien, Materialien  |
|---|---|---|
| <p><b>1. Nutzung von relationalen Datenbanken</b></p> <p>(a) Aufbau von Datenbanken und Grundbegriffe</p> <ul style="list-style-type: none"> <li>Entwicklung von Fragestellungen zur vorhandenen Datenbank</li> <li>Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, Datenbankschema</li> </ul> <p>(b) SQL-Abfragen</p> <ul style="list-style-type: none"> <li>Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL (<code>SELECT (DISTINCT) ...FROM, WHERE, AND, OR, NOT</code>) auf einer Tabelle</li> <li>Analyse und Erarbeitung von SQL-Abfragen auf einer und mehrerer Tabelle zur Beantwortung der Fragestellungen (<code>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</code>)</li> </ul> <p>(c) Vertiefung an einem weiteren Datenbankbeispiel</p> | <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A),</li> <li>analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A),</li> <li>analysieren und erläutern eine Datenbankmodellierung (A),</li> <li>erläutern die Eigenschaften normalisierter Datenbankschemata (A),</li> <li>bestimmen Primär- und Sekundärschlüssel (M),</li> <li>ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M),</li> <li>modifizieren eine Datenbankmodellierung (M),</li> <li>modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M),</li> <li>bestimmen Primär- und Sekundärschlüssel (M),</li> <li>überführen Datenbankschemata in vorgegebene Normalformen (M),</li> <li>verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I),</li> </ul> | <p><i>Beispiel: Fahrradgeschäft</i><br/>Ein Fahrradgeschäft will seine Fahrräder in einer Datenbank verwalten. Die Schülerinnen und Schüler analysieren die Aufgabe und setzen diese mit Hilfe eines Schemas, seinen Attributen usw. um.</p> <p><i>Beispiel: Mensaessen</i><br/>Die Kombination mehrere Tabellen und der anschließenden Normalisierung erfolgt zum Beispiel durch die Kombination von Vorspeisen, Hauptgerichten und Nachspeisen einer Mensa.</p> <p><i>Beispiel: Buchhandlung</i><br/>Eine Buchhandlung möchte seine Bücher in einer Datenbank speichern. Dabei sollen keine Redundanzen und Anomalien entstehen, deshalb muss die Datenbank in die 3 Normalform gebracht werden.</p> <p><i>Beispiel: Buchungssystem</i><br/>In dem Online-Buchungssystem einer Schule können die Lehrer Medienräume, Beamer, Laptops, Kameras, usw. für einen bestimmten Zeitpunkt buchen, der durch Datum und die Schulstunde festgelegt ist. Dazu ist die Datenbank zu modellieren, ggf. zu normalisieren und im Datenbanksystem umzusetzen. Weiter sollen sinnvolle Abfragen entwickelt werden.<br/>Unter <a href="http://mrbs.sourceforge.net">http://mrbs.sourceforge.net</a> (abgerufen: 30.03. 2014) findet man ein freies Online-Buchungssystem inklusive Demo, an Hand derer man erläutern kann, worum es in dem Projekt geht.</p> |
| <p><b>2. Modellierung von relationalen Datenbanken</b></p>  | <ul style="list-style-type: none"> <li>ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D),</li> </ul>  |   |

|   |   |  |
|---|---|--|
| <p>(a) Entity-Relationship-Diagramm</p> <ul style="list-style-type: none"> <li>• Ermittlung von Entitäten, zugehörigen Attributen, Relationen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms</li> <li>• Erläuterung und Modifizierung einer Datenbankmodellierung</li> </ul> <p>(b) Entwicklung einer Datenbank aus einem Datenbankentwurf</p> <ul style="list-style-type: none"> <li>• Modellierung eines relationalen Datenbankschemas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln</li> </ul> <p>(c) Redundanz, Konsistenz und Normalformen</p> <ul style="list-style-type: none"> <li>• Untersuchung einer Datenbank hinsichtlich Konsistenz und Redundanz in einer Anwendungssituation</li> <li>• Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten)</li> </ul> | <ul style="list-style-type: none"> <li>• stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D),</li> <li>• überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D).</li> </ul> | <p><i>Beispiel: Schulverwaltung</i></p> <p>In einer Software werden die Schulhalbjahre, Jahrgangsstufen, Kurse, Klassen, Schüler, Lehrer und Noten einer Schule verwaltet. Man kann dann ablesen, dass z.B. Schüler X von Lehrer Y im 2. Halbjahr des Schuljahrs 2011/2012 in der Jahrgangsstufe 9 im Differenzierungsbereich im Fach Informatik die Note „sehr gut“ erhalten hat. Dazu ist die Datenbank zu modellieren, ggf. zu normalisieren und im Datenbanksystem umzusetzen. Weiter sollen sinnvolle Abfragen entwickelt werden und das Thema Datenschutz besprochen werden.</p> |
|---|---|--|



## Unterrichtsvorhaben Q1-VII:

**Thema:** Sicherheit und Datenschutz in Netzstrukturen

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

### Vorhabenbezogene Konkretisierung:

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.

**Zeitbedarf:** 6 Stunden

### Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen   | Zu entwickelnde Kompetenzen   | Beispiele, Medien, Materialien  |
|--|---|---|
| <b>1. Daten in Netzwerken und Sicherheitsaspekte in Netzen sowie beim Zugriff auf Datenbanken</b><br>(a) Beschreibung eines Datenbankzugriffs im Netz anhand eines Anwendungskontextes und einer Client-Server-Struktur zur Klärung der Funktionsweise eines Datenbankzugriffs<br>(b) Netztopologien als Grundlage von Client-Server-Strukturen und TCP/IP-Schichtenmodell als Beispiel für eine Paketübermittlung in einem Netz | Die Schülerinnen und Schüler <ul style="list-style-type: none"><li>• beschreiben und erläutern Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A),</li><li>• analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A),</li><li>• 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),</li><li>• untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatik-</li></ul> | <i>Materialien:</i><br>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben, Verschlüsselung Q1.5 - Zugriff auf Daten in Netzwerken              |
| <b>2. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht</b>  |   | <i>Materialien:</i><br>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1 5 - Datenschutz beim Videocenter, Materialblatt-Datenschutzgesetz |

---

|  |   |  |
|--|---|--|
|  | <p>systemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A),</p> <ul style="list-style-type: none"><li>• nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D).</li></ul> |  |
|--|---|--|

---

## Unterrichtsvorhaben Q2-I:

**Thema:** Endliche Automaten und formale Sprachen

**Leitfragen:** *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?*

### **Vorhabenbezogene Konkretisierung:**

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.

Einige Automaten werden an der Programmierumgebung Kara dargestellt, analysiert und bearbeitet. Anschließend werden die Schülerinnen und Schüler mit Hilfe von Kara den Begriff der Nebenläufigkeit kennenlernen und damit Arbeiten.

**Zeitbedarf:** 24 Stunden

## Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen   | Zu entwickelnde Kompetenzen   | Beispiele, Medien oder Materialien   |
|--|---|--|
| <p><b>1. Endliche Automaten</b></p> <p>(a) Vom Automaten in den Schülerinnen und Schülern bekannten Kontexten zur formalen Beschreibung eines endlichen Automaten</p> <p>(b) Untersuchung, Darstellung und Entwicklung endlicher Automaten</p> <p>(c) Endlichen Automaten am Beispiel von AutomatenKara</p>  | <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens auf bestimmte Eingaben (A),</li> <li>analysieren und erläutern Grammatiken regulärer Sprachen (A),</li> <li>zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A),</li> <li>ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A),</li> <li>entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M),</li> <li>entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M),</li> <li>entwickeln zur akzeptierten Sprache eines Automaten die zugehörige Grammatik (M),</li> <li>entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M),</li> <li>modifizieren Grammatiken regulärer Sprachen (M),</li> <li>entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M),</li> <li>stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in</li> </ul> | <p><i>Beispiele:</i><br/>Cola-Automat, Geldspielautomat, Roboter, Zustandsänderung eines Objekts „Auto“, Akzeptor für bestimmte Zahlen, Akzeptor für Teilwörter in längeren Zeichenketten, Akzeptor für Terme</p> <p><i>Beispiele:</i><br/>Kara der Marienkäfer löst Aufgaben in seiner Welt mit Hilfe von Automaten</p> |
| <p><b>2. Untersuchung und Entwicklung von Grammatiken regulärer Sprachen</b></p> <p>(a) Erarbeitung der formalen Darstellung regulärer Grammatiken</p> <p>(b) Untersuchung, Modifikation und Entwicklung von Grammatiken</p> <p>(c) Entwicklung von endlichen Automaten zum Erkennen regulärer Sprachen die durch Grammatiken gegeben werden</p> <p>(d) Entwicklung regulärer Grammatiken zu endlichen Automaten</p> | <ul style="list-style-type: none"> <li>entwickeln zur akzeptierten Sprache eines Automaten die zugehörige Grammatik (M),</li> <li>entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M),</li> <li>modifizieren Grammatiken regulärer Sprachen (M),</li> <li>entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M),</li> <li>stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in</li> </ul>   | <p><i>Beispiele:</i><br/>reguläre Grammatik für Wörter mit ungerader Parität, Grammatik für Wörter, die bestimmte Zahlen repräsentieren, Satzgliederungsgrammatik</p> <p><i>Beispiele:</i><br/><i>Autonummernschilder entsprechen immer einer formalen Sprache</i></p>   |
| <p><b>3. Grenzen endlicher Automaten</b></p>   | <ul style="list-style-type: none"> <li>stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in</li> </ul>   | <p><i>Beispiele:</i><br/>Klammerausdrücke, <math>a^n b^n</math> im Vergleich zu <math>(ab)^n</math></p>  |

|  |  |   |
|--|--|---|
|  | <p>die jeweils andere Darstellungsform (D),</p> <ul style="list-style-type: none"> <li>• ermitteln die Sprache, die ein endlicher Automat akzeptiert (D).</li> <li>• beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D).</li> </ul> |   |
| <p><b>4. Nebenläufige Prozesse</b><br/>Anhand von MultiKara erlernen die Schülerinnen und Schüler das Prinzip der Nebenläufigkeit kennen</p> |  | <p><i>Beispiele:</i><br/>Der Shedular bei MultiKara bestimmt, welche Kara an der Reihe ist. Andere Verfahren werden besprochen.</p> |

## Unterrichtsvorhaben Q2-II:

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

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

### Vorhabenbezogene Konkretisierung:

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

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

**Zeitbedarf:** 12 Stunden

### Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen  | Zu entwickelnde Kompetenzen  | Beispiele, Medien oder Materialien  |
|---|--|---|
| <p><b>1. Von-Neumann-Architektur und die Ausführung maschinennaher Programme</b></p> <p>a) prinzipieller Aufbau einer von Neumann-Architektur mit CPU, Rechenwerk, Steuerwerk, Register und Hauptspeicher</p> <p>b) einige maschinennahe Befehle und ihre Repräsentation in einem Binär-Code, der in einem Register gespeichert werden kann</p> <p>c) Analyse und Erläuterung der Funkti-</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> | <p><i>Beispiel:</i><br/>Addition von 4 zu einer eingegebenen Zahl mit einem Rechnermodell</p> <p><i>Materialien:</i><br/>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 –Von-Neumann-Architektur und maschinennahe Programmierung (<a href="#">Download Q2-III.1</a>)</p> |

|  |  |   |
|--|--|---|
| <p>onsweise eines einfachen maschinennahen Programms</p>   |  |   |
| <p><b>2. Grenzen der Automatisierbarkeit</b></p> <ul style="list-style-type: none"> <li>a) Vorstellung des Halteproblems</li> <li>b) Unlösbarkeit des Halteproblems</li> <li>c) Beurteilung des Einsatzes von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen</li> </ul> |  | <p><i>Beispiel:</i> Halteproblem</p> <p><i>Materialien:</i><br/>Ergänzungsmaterialien zum Lehrplannavigator<br/>Unterrichtsvorhaben Q2.3 - Halteproblem<br/>(<a href="#">Download Q2-III.2</a>)</p> |

## Unterrichtsvorhaben Q2-III:

**Thema:** Verschlüsselung bei der Kommunikation sowie der zugehörige Datenschutz

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

### **Vorhabenbezogene Konkretisierung:**

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

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

**Zeitbedarf:** 12 Stunden

### **Sequenzierung des Unterrichtsvorhabens:**

| <b>Unterrichtssequenzen</b>   | <b>Zu entwickelnde Kompetenzen</b>   | <b>Beispiele, Medien oder Materialien</b> |
|---|--|---|
| <p><b>1. Daten in Netzwerken und Sicherheitsaspekte in Netzen sowie beim Zugriff auf Datenbanken</b></p> <p>(a) Beschreibung eines Datenbankzugriffs im Netz anhand eines Anwendungskontextes und einer Client-Server-Struktur zur Klärung der Funktionsweise eines Datenbankzugriffs</p> <p>(b) Netztopologien als Grundlage von Client-Server-Strukturen und TCP/IP-Schichtenmodell als Beispiel für eine</p> | <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"><li>• beschreiben und erläutern Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A),</li><li>• analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A),</li><li>• untersuchen und bewerten anhand von Fallbeispielen die Auswirkungen des</li></ul> |   |



|  |  |  |
|--|--|--|
| <p>Paketübermittlung in einem Netz<br/>(c) Vertraulichkeit, Integrität, Authentizität in Netzwerken sowie symmetrische und asymmetrische kryptografische Verfahren (Cäsar-, Vigenère-, RSA-Verfahren) als Methoden Daten im Netz verschlüsselt zu übertragen</p> | <p>Einsatzes von Informatiksystemen, die Sicherheit von Informatiksystemen sowie die Einhaltung der Datenschutzbestimmungen und des Urheberrechts (A),</p> <ul style="list-style-type: none"> <li>• untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A),</li> </ul> |  |
| <p><b>2. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht</b></p>   | <ul style="list-style-type: none"> <li>• nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D).</li> </ul>   |  |

---

## **Unterrichtsvorhaben Q2-IV:**

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

---

## 2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

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

### Überfachliche Grundsätze:

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

### Fachliche Grundsätze:

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

---

## 2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

**Hinweis:** Sowohl die Schaffung von **Transparenz bei Bewertungen** als auch die Vergleichbarkeit von Leistungen sind das Ziel, innerhalb der gegebenen Freiräume Vereinbarungen zu Bewertungskriterien und deren Gewichtung zu treffen.

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

### 2.3.1 Beurteilungsbereich Klausuren

#### **Verbindliche Absprachen:**

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

#### **Instrumente:**

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

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

#### **Kriterien**

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

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

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

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

---

### 2.3.2 Beurteilungsbereich Sonstige Mitarbeit

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

#### Verbindliche Absprachen der Fachkonferenz

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

#### Leistungsaspekte

##### Mündliche Leistungen

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

##### Praktische Leistungen am Computer

- Implementierung, Test und Anwendung von Informatiksystemen

##### Sonstige schriftliche Leistungen

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

#### Kriterien

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

Die Bewertungskriterien stützen sich auf

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

Besonderes Augenmerk ist dabei auf

- die sachliche Richtigkeit,
- die angemessene Verwendung der Fachsprache,

- 
- die Darstellungskompetenz,
  - die Komplexität und den Grad der Abstraktion,
  - die Selbstständigkeit im Arbeitsprozess,
  - die Präzision und
  - die Differenziertheit der Reflexion zu legen.

Bei Gruppenarbeiten auch auf

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

Bei Projektarbeit darüber hinaus auf

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

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

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

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

Die Leistungsrückmeldung kann

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

erfolgen.

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

---

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

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

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

Im Informatikunterricht werden Kompetenzen anhand informatischer Inhalte in verschiedenen Anwendungskontexten erworben, in denen Schülerinnen und Schülern aus anderen Fächern Kenntnisse mitbringen können. Diese können insbesondere bei der Auswahl und Bearbeitung von Softwareprojekten berücksichtigt werden und in einem hinsichtlich der informatischen Problemstellung angemessenem Maß in den Unterricht Eingang finden. Da im Inhaltsfeld Informatik, Mensch und Gesellschaft auch gesellschaftliche und ethische Fragen im Unterricht angesprochen werden, soll eine mögliche Zusammenarbeit mit dem Fach Sozialwissenschaften angestrebt werden.

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

Möglichst schon im zweiten Halbjahr der Einführungsphase, spätestens jedoch im ersten Halbjahr des ersten Jahres der Qualifikationsphase werden im Unterricht an geeigneten Stellen Hinweise zur Erstellung von Facharbeiten gegeben. Es wird vereinbart, dass nur Facharbeiten vergeben werden, die mit der eigenständigen Entwicklung eines Softwareproduktes verbunden sind oder eine Neuzusammenstellung eines aktuellen Themas beinhaltet.

---

## 4 Qualitätssicherung und Evaluation

Das schulinterne Curriculum stellt keine starre Größe dar, sondern ist als „lebendes Dokument“ zu betrachten. Dementsprechend sind die Inhalte stetig zu überprüfen, um ggf. Modifikationen vornehmen zu können. Die Fachkonferenz (als professionelle Lerngemeinschaft) trägt durch diesen Prozess zur Qualitätsentwicklung und damit zur Qualitätssicherung des Faches bei.

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

Das schulinterne Curriculum (siehe 2.1) ist zunächst bis 2017 für den ersten Durchgang durch die gymnasiale Oberstufe nach Erlass des Kernlehrplanes verbindlich. 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.