



St.-Franziskus-Gymnasium

*Staatlich genehmigtes privates Gymnasium
für Jungen und Mädchen*



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

Fach

Informatik

Stand: 02.12.2017

Inhalt

	Seite
1 Das Fach Informatik am St.-Franziskus-Gymnasium	3
2 Entscheidungen zum Unterricht	5
2.1 Unterrichtsvorhaben	5
2.1.1 Übersichtsraster Unterrichtsvorhaben	6
2.1.2 Konkretisierte Unterrichtsvorhaben	9
2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit	41
2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung	42
2.3.1 Grundsätze der Leistungsbewertung	42
2.3.2 Formen der Leistungsüberprüfung	42
2.3.3 Grundsätze der Leistungsrückmeldung und Beratung	44
2.3.4 Individuelle Förderung	45
2.3.5 Bildung der Zeugnisnote	45
3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen	45
4 Qualitätssicherung und Evaluation	46

1 Das Fach Informatik am St.-Franziskus-Gymnasium Olpe

Das Fach Informatik wird am St.-Franziskus-Gymnasium in der gymnasialen Oberstufe als Grundkurs und in der Mittelstufe im Rahmen der Differenzierung in Kombination mit dem Fach Mathematik in den Klassen 8 und 9 angeboten.

Daneben erhalten alle Schülerinnen der Jahrgangsstufe 5 eine Mediens Schulung. Kinder und Jugendliche haben Medien ganz selbstverständlich in ihren Alltag integriert. Im Umgang damit erwerben sie wichtige Schlüsselkompetenzen zur Teilhabe in unserer Gesellschaft. Je bedeutender Medien im Alltag der Kinder und Jugendlichen werden, umso mehr steigen auch die Anforderungen an ihre Medienkompetenz. Ziel der Mediens Schulung ist die Vermittlung grundlegender Kompetenzen im Umgang mit den neuen Medien (u.a. Internet, What's App, Suchmaschinen, Cyber-Mobbing, Datenschutz und Sicherheit, ...).

Des Weiteren besteht noch die Möglichkeit, in der Jahrgangsstufe 5 bzw. 6 an einer Wahlpflicht-AG PC teilzunehmen. Im Rahmen dieser AG erlernen die Schülerinnen und Schüler grundlegende Techniken im Umgang mit der Textverarbeitung. Daneben erhalten sie eine Einführung in die Bildschirmpräsentation und in die Tabellenkalkulation. Zudem erfahren die Schülerinnen und Schüler, wie eine sinnvolle Internetrecherche aussehen sollte. Dabei lernen sie u.a. verschiedene Suchmaschinen und deren Arbeitsweisen kennen.

Im Rahmen von Mediens Schulungen finden zudem in den Jahrgangsstufen 7 bis 9 und in der EF u.a. in der Fahrtenwoche ein- bis zweistündige Einführungen bzw. Vertiefungen in die verschiedenen Standardapplikationen (Textverarbeitung, Bildschirmpräsentation und Tabellenkalkulation) statt. Zudem wird im Fach Mathematik verstärkt auf die Tabellenkalkulation eingegangen.

Der Differenzierungskurs Informatik/Mathematik in den Jahrgangsstufen 8 und 9 vereint Inhalte und Methoden der Fächer Informatik und Mathematik miteinander. Der Schwerpunkt liegt aber eindeutig auf der Informatik. Neben einer fundierten Einführung bzw. Vertiefung in die Textverarbeitung und in die Tabellenkalkulation erhalten die Schülerinnen und Schüler hier eine Einführung in die Programmierung; zunächst mit Hilfe des Programmiersystems Robot Karol und anschließend mittels der einfachen imperativen Programmiersprache Logo.

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

Der Unterricht der Sekundarstufe II wird mit Hilfe der objektorientierten Programmiersprache Java unter der Entwicklungsumgebung BlueJ durchgeführt. In der Einführungsphase und zum Teil auch in der Qualifikationsphase kommt dabei zusätzlich die didaktische Stifte-und-Mäuse-Bibliothek zum Einsatz, welche das Erstellen von Programmen erleichtert.

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

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

Zurzeit besteht die Fachschaft Informatik des St.-Franziskus-Gymnasiums aus drei Lehrkräften, denen zwei Computerräume mit jeweils 16 Schülerarbeitsplätzen zur Verfügung stehen. Alle Arbeitsplätze sind an das schulinterne Rechnernetz angeschlossen, so dass Schülerinnen und Schüler über einen Zugang zum zentralen Schulserver alle Arbeitsplätze der zwei Computerräume sowie die 10 frei zugänglichen Arbeitsplätze in der Schülerbibliothek zum Zugriff auf ihre eigenen Daten, zur Recherche im Internet bzw. zur Bearbeitung schulischer Aufgaben verwenden können.

Der Unterricht erfolgt am St.-Franziskus-Gymnasium im 67,5-Minuten-Takt. Die Unterrichtsverteilung sieht für Grundkurse in der gymnasialen Oberstufe grundsätzlich zwei Einzelstunden pro Woche vor.

2 Entscheidungen zum Unterricht

2.1 Unterrichtsvorhaben

Im Fach Informatik geht es im Wesentlichen darum, zu einer gegebenen informatischen Problemstellung ein effizientes Lösungsverfahren zu entwickeln, das dann mit Hilfe eines Computers umgesetzt werden kann. Der Unterricht orientiert sich dabei an den [fundamentalen Ideen der Informatik](#) nach Andreas Schwill (veröffentlicht im Zentralblatt für Didaktik der Mathematik 1 (1993), S. 20-31 der Universität Potsdam),

- Algorithmisierung,
- Strukturierte Zerlegung,
- Sprache.

Dabei kommt es nicht auf kurzfristig anwendbares, abfragbares Wissen sondern vielmehr auf grundlegende Verfahren und Konzepte (Kompetenzen) an, die längerfristig Gültigkeit besitzen.

Mit Beginn des Unterrichts in der Einführungsphase wird die Programmiersprache *Java* gelehrt. Da es sich für viele Schülerinnen und Schüler um ein neu einsetzendes Fach handelt, sind Vorkenntnisse in diesem Bereich nicht nötig.

Die im Folgenden dargestellten Unterrichtsvorhaben sind auf Grundlage des [Kernlehrplans Informatik für die Sekundarstufe II Gymnasium / Gesamtschule in NRW](#) erstellt worden.

Ein Unterrichtsvorhaben setzt sich aus **Kompetenzbereichen** und **Inhaltsfeldern** zusammen. Diese werden im Kernlehrplan ausführlich beschrieben. Hier eine Übersicht:

Kompetenzbereiche:

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

Inhaltsfelder:

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

Die Darstellung der Unterrichtsvorhaben im schulinternen Lehrplan besitzt den Anspruch, sämtliche im Kernlehrplan aufgeführten Kompetenzen abzudecken. Dies bedeutet für die 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, die im Folgenden beschrieben werden.

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, aktuellen Themen bzw. die Erfordernisse anderer besonderer Ereignisse (z. B. Exkursionen, Praktika, 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 bindend sind, besitzt die exemplarische Ausweisung „konkretisierter Unterrichtsvorhaben“ (Kapitel 2.1.2) empfehlenden Charakter. 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, fachübergreifenden Kooperationen, Lernmitteln und Lernorten sowie vorgesehenen Leistungsüberprüfungen, die im Einzelnen auch den Kapiteln 2.2 bis 2.3 zu entnehmen sind. Abweichungen von den vorgeschlagenen Vorgehensweisen bezüglich der konkretisierten Unterrichtsvorhaben sind im Rahmen der pädagogischen Freiheit der Lehrkräfte jederzeit möglich. Sicherzustellen bleibt allerdings auch hier, dass im Rahmen der Umsetzung der Unterrichtsvorhaben insgesamt alle konkretisierten Kompetenzerwartungen des Kernlehrplans Berücksichtigung finden.

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>Thema: <i>Überblick über die Informatik, Einführung in die Grundlagen und die Nutzung von Informatiksystemen</i></p> <p>Zentrale Kompetenzen: Argumentieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Informatiksysteme Informatik, Mensch und Gesellschaft</p> <p>Inhaltliche Schwerpunkte: Einsatz, Nutzung und Aufbau von Informatiksystemen Binäre Codierung und Verarbeitung Besondere Eigenschaften der digitalen Speicherung und Verarbeitung von Daten</p> <p>Zeitbedarf: 6 67,5-Std. (≙ 9 45-Std.)</p>	<p><u>Unterrichtsvorhaben E-II</u></p> <p>Thema: <i>Grundlagen der objektorientierten Analyse, Modellierung und Implementierung</i></p> <p>Zentrale Kompetenzen: Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Daten und ihre Strukturierung Formale Sprachen und Automaten</p> <p>Inhaltliche Schwerpunkte: Objekte und Klassen Syntax und Semantik einer Programmiersprache (hier Java)</p> <p>Zeitbedarf: 6 67,5-Std. (≙ 9 45-Std.)</p>

Einführungsphase	
<p><u>Unterrichtsvorhaben E-III</u></p> <p>Thema: <i>Algorithmische Grundstrukturen in Java</i></p> <p>Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Algorithmen Daten und ihre Strukturierung Formale Sprachen und Automaten</p> <p>Inhaltliche Schwerpunkte: Objekte und Klassen Syntax und Semantik einer Programmiersprache Analyse, Entwurf und Implementierung einfacher Algorithmen</p> <p>Zeitbedarf: 10 67,5-Std (≙ 15 45-Std.)</p>	<p><u>Unterrichtsvorhaben E-IV</u></p> <p>Thema: <i>Modellierung und Implementierung von Klassen und deren Beziehungen</i></p> <p>Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Algorithmen Daten und ihre Strukturierung Formale Sprachen und Automaten</p> <p>Inhaltliche Schwerpunkte: Objekte und Klassen Syntax und Semantik einer Programmiersprache Analyse, Entwurf und Implementierung einfacher Algorithmen</p> <p>Zeitbedarf: 16 67,5-Std. (≙ 24 45-Std.)</p>

Einführungsphase	
<p><u>Unterrichtsvorhaben E-V</u></p> <p>Thema: <i>Einführung in Such- und Sortieralgorithmen auf Feldern anhand kontextbezogener Beispiele</i></p> <p>Zentrale Kompetenzen: Argumentieren Modellieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Algorithmen Daten und ihre Strukturierung</p> <p>Inhaltliche Schwerpunkte: Algorithmen zum Suchen und Sortieren Analyse und Entwurf einfacher Algorithmen</p> <p>Zeitbedarf: 10 67,5-Std. (≙ 15 45-Std.)</p>	<p><u>Unterrichtsvorhaben E-VI</u></p> <p>Thema: <i>Geschichte der digitalen Datenverarbeitung und Grundlagen des Datenschutzes</i></p> <p>Zentrale Kompetenzen: Argumentieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Informatik, Mensch und Gesellschaft Informatiksysteme</p> <p>Inhaltliche Schwerpunkte: Geschichte der automatischen Datenverarbeitung Wirkungen der Automatisierung</p> <p>Zeitbedarf: 6 67,5-Std. (≙ 9 45-Std.)</p>
Summe Einführungsphase: 54 67,5-Std. (81 45-Std.)	

II. Qualifikationsphase (Q1 und Q2) - GRUNDKURS

Qualifikationsphase 1	
<p><u>Unterrichtsvorhaben Q1-I</u></p> <p>Thema: <i>Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Daten und ihre Strukturierung • Algorithmen • Informatiksysteme <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Objekte und Klassen • Analyse, Entwurf und Implementierung von Algorithmen • Syntax und Semantik einer Programmiersprache • Nutzung von Informatiksystemen <p>Zeitbedarf: 10 67,5-Std. (≙ 15 45-Std.)</p>	<p><u>Unterrichtsvorhaben Q1-II</u></p> <p>Thema: <i>Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Daten und ihre Strukturierung • Algorithmen • Informatiksysteme <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Objekte und Klassen • Analyse, Entwurf und Implementierung von Algorithmen • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache • Nutzung von Informatiksystemen <p>Zeitbedarf: 16 67,5-Std. (≙ 24 45-Std.)</p>

Qualifikationsphase 1	
<p><u>Unterrichtsvorhaben Q1-III</u></p> <p>Thema: <i>Suchen und Sortieren auf linearen Datenstrukturen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Algorithmen • Informatiksysteme <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Analyse, Entwurf und Implementierung von Algorithmen • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache • Nutzung von Informatiksystemen <p>Zeitbedarf: 14 67,5-Std. (≙ 21 45-Std.)</p>	<p><u>Unterrichtsvorhaben Q1-IV</u></p> <p>Thema: <i>Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Objekte und Klassen • Analyse, Entwurf und Implementierung von Algorithmen • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: 16 67,5-Std. (≙ 24 45-Std.)</p>
Summe Qualifikationsphase 1: 56 67,5-Std. (84 45-Std.)	

Qualifikationsphase 2	
<p><u>Unterrichtsvorhaben Q2-I</u></p> <p>Thema: <i>Endliche Automaten und formale Sprachen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Modellieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Endliche Automaten und formale Sprachen <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Endliche Automaten • Grammatiken regulärer Sprachen • Möglichkeiten und Grenzen von Automaten und formalen Sprachen <p>Zeitbedarf: 14 67,5-Std. (≙ 21 45-Std.)</p>	<p><u>Unterrichtsvorhaben Q2-II</u></p> <p>Thema: <i>Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten • Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Datenbanken • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache • Sicherheit <p>Zeitbedarf: 16 67,5-Std. (≙ 24 45-Std.)</p>

Qualifikationsphase 2	
<p><u>Unterrichtsvorhaben Q2-III</u></p> <p>Thema: <i>Aufbau von und Kommunikation in Netzwerken</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Informatiksysteme • Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Einzelrechner und Rechnernetzwerke • Nutzung von Informatiksystemen • Wirkungen der Automatisierung <p>Zeitbedarf: 8 67,5-Std. (≙ 12 45-Std.)</p>	<p><u>Unterrichtsvorhaben Q2-IV</u></p> <p>Wiederholung und Vertiefung ausgewählter Kompetenzen und Inhalte des ersten Jahrs der Qualifikationsphase</p>
Summe Qualifikationsphase 2: 38 67,5-Std. (57 45-Std.)	

2.1.2 Konkretisierte Unterrichtsvorhaben

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

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

In der Einführungsphase wird zusätzlich die didaktische Bibliothek *Stifte-und-Mäuse* verwendet. Diese Bibliothek wird den Schülerinnen und Schülern im Rahmen eines Installationspaketes zur Verfügung gestellt. Zur Implementierung von Java-Programmen bzw. Projekten wird der Editor *BlueJ* verwendet, der kostenfrei aus dem Internet heruntergeladen werden kann.

In der Qualifikationsphase werden die Unterrichtsvorhaben unter Berücksichtigung der Vorgaben für das Zentralabitur Informatik in NRW konkretisiert. Diese sind zu beziehen unter <http://www.standardsicherung.schulministerium.nrw.de/abitur-gost/fach.php?fach=15>.

I Einführungsphase

Die folgenden Kompetenzen aus dem Bereich *Kommunizieren und Kooperieren* (K) werden in allen Unterrichtsvorhaben 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: Überblick über die Informatik, Einführung in die Grundlagen und die Nutzung von Informatiksystemen

Leitfragen: *Womit beschäftigt sich die Wissenschaft der Informatik? Mit welchen Themen befasst sich das Fach Informatik in der Schule? Wie funktioniert ein moderner Computer? 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 Faches behandelt werden sollten.

Zunächst wird auf den zentralen Begriff der Information eingegangen und die Möglichkeit der Codierung von Daten thematisiert. Insbesondere wird dabei auf die Binärdarstellung von Zahlen eingegangen.

Des Weiteren soll der grundlegende Aufbau eines Rechnersystems im Sinne der Von-Neumann-Architektur erarbeitet werden und mit dem grundlegenden Prinzip der Datenverarbeitung (Eingabe-Verarbeitung-Ausgabe, EVA-Prinzip) in Beziehung gesetzt werden.

Außerdem werden die Schülerinnen und Schüler in die konkrete Nutzung der Informatiksysteme an der Schule eingewiesen.

Zeitbedarf: 6 67,5-Std. (≙ 9 45-Std.)

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Allgemeine Einführung a) Übersicht über das Fach b) Einführung in die Informatiksysteme der Schule	Die Schülerinnen und Schüler <ul style="list-style-type: none"> • nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D), • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K). 	Computerräume der Schule
2. Grundlagen von Informatiksystemen a) Darstellung von Zahlen im Binärsystem b) Von-Neumann-Architektur c) EVA-Prinzip	Die Schülerinnen und Schüler <ul style="list-style-type: none"> • stellen ganze Zahlen und Zeichen in Binärcodes dar (D) • interpretieren Binärcodes als Zahlen und Zeichen (D) • beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A). 	Demonstrationshardware

Unterrichtsvorhaben EF-II

Thema: Grundlagen der objektorientierten Analyse, Modellierung und Implementierung

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

Vorhabenbezogene Konkretisierung:

Ein zentraler Bestandteil des Informatikunterrichts der Einführungsphase ist die Objekt-orientierte 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.

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 Objekt- und Klassenkarten oder Beziehungsdiagramme eingeführt.

Im Anschluss wird mit der Realisierung erster kleiner Programme in Java unter der Entwicklungsumgebung BlueJ mit Hilfe der didaktischen Stifte-und-Mäuse-Klassenbibliothek begonnen. Die von der Kernbibliothek dieser Bibliothek vorgegebenen Klassen (u.a. Bildschirm, Maus, Stift und Tastatur) werden mit den Schülerinnen und Schülern in Teilen analysiert und entsprechende Objekte anhand einfacher Problemstellungen erstellt. 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.

Zeitbedarf: 6 67,5-Std. (≙ 9 45-Std.)

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Identifikation von Objekten</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) Typengleiche Objekte werden so zu einer Objektsorte bzw. Objektklasse zusammengefasst.</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen (Methoden) und ihre Beziehungen (M) modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M) 	<p><i>Beispiel:</i> Tiere</p> <p>Die Schülerinnen und Schüler betrachten verschiedene Tiere und klassifizieren diese über ihre Attribute (Eigenschaften) und Methoden (Dienste)</p>

<p>2. Analyse von Kernklassen der didaktische Klassenbibliothek</p> <p>a) Objektorientierte Programmierung als modularisiertes Vorgehen (Entwicklung von Problemlösungen auf der Grundlage vorhandener Klassen)</p> <p>b) Analyse der Kernklassen der didaktischen Klassenbibliothek SuM (Klassen des Bibliothekpakets sum.kern)</p>	<ul style="list-style-type: none"> • stellen die Beziehungen zwischen Klassen grafisch dar (M) • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I) • interpretieren Fehlermeldungen (D) und korrigieren den Quellcode (I) 	<p><i>Medien/Materialien:</i> Programmiersprache Java, Entwicklungsumgebung BlueJ</p>
<p>3. Implementierung einfacher Programme</p> <p>a) Grundaufbau einer Java-Klasse</p> <p>b) Deklaration und Initialisierung von Objekten</p> <p>c) Methodenaufrufe mit Parameterübergabe zur Manipulation von Objekteigenschaften (z.B. Position, Drehung)</p>		<p><i>Projekte:</i></p> <ul style="list-style-type: none"> • MeinErstesProgramm • Traumhaus • Figuren (programmgesteuertes Zeichnen)

Unterrichtsvorhaben EF-III

Thema: Algorithmische Grundstrukturen in Java

Leitfragen: *Wie lassen sich einfache Animationen und Simulationen optischer Gegenstandsbereiche unter Berücksichtigung von Maus- und Tastatureingaben steuern?*

Vorhabenbezogene Konkretisierung:

Der Schwerpunkt dieses Unterrichtsvorhabens liegt auf der Entwicklung von Projekten, die durch Eingaben des Benutzers gesteuerte Animationen aufweisen.

Zunächst wird stufenweise ein Projekt erarbeitet, bei dem durch Mausbewegungen mit Hilfe eines Stiftes auf dem Bildschirm gezeichnet werden kann. Mit Hilfe der Tastatur soll zusätzlich die Funktionalität des Stiftes (Zeichen- bzw. Radiermodus) bzw. die Zeichenfarbe geändert werden können. Für die Umsetzung dieses Projekts werden Kontrollstrukturen in Form von Schleifen und Verzweigungen benötigt. Die Aufgabenstellung wird dabei mehrfach abgewandelt, um die verschiedenen Kontrollstrukturen einzuführen. Zur grafischen Darstellung der Kontrollstrukturen werden Struktogramme eingeführt.

Daran anschließend sollen in einem weiteren lebensnahen Projekt (interaktives Dartspiel) die eingeführten Kontrollstrukturen eingeübt und gefestigt werden.

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

Zeitbedarf: 10 67,5-Std. (≙ 15 45-Std.)

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Erzeugung eines Malprogramms</p> <p>a) Schrittweise Einführung der verschiedenen Kontrollstrukturen (Schleifen und Verzweigungen)</p> <p>b) Einführung von Struktogrammen zur grafischen Darstellung von Kontrollstrukturen</p> <p>c) IST-Beziehung (Unterscheidung zwischen Ober- und Unterklasse)</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern einfache Algorithmen und Programme (A) • entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M) • implementieren Algorithmen unter Verwendung von Kontrollstrukturen sowie Methodenaufrufen (I) 	<p>Programmiersprache Java, Entwicklungsumgebung BlueJ</p> <p>Struktogrammeditor zur Erzeugung von Nassi-Shneiderman Diagrammen</p>
<p>2. Animation grafischer Objekte</p> <p>a) Anwendung verschiedener Kontrollstrukturen</p> <p>b) Analyse und Verwendung der Klasse Rechner aus dem Bibliothekspaket sum.werkzeuge</p> <p>c) Verknüpfung von Bedingungen durch die logische Verknüpfungen UND, ODER und NICHT und deren Umsetzung in Java</p>	<ul style="list-style-type: none"> • modifizieren einfache Algorithmen und Programme (I) • interpretieren Fehlermeldungen (D) und korrigieren den Quellcode (I). 	<p><i>Projekte:</i></p> <ul style="list-style-type: none"> • Freihandzeichnen • Peilwurf (Dartspiel)

Unterrichtsvorhaben EF-IV

Thema: Modellierung und Implementierung von Klassen und deren Beziehungen

Leitfragen: *Wie werden realistische Systeme anforderungsspezifisch reduziert, modelliert und implementiert? Wie stehen Klassen in Beziehung zueinander und wie wird dieses dargestellt und realisiert?*

Vorhabenbezogene Konkretisierung:

Das Unterrichtsvorhaben hat die Entwicklung und Dokumentation eigener Klassen und ihre Beziehungen untereinander zum Schwerpunkt.

Zunächst werden ausgehend von der bekannten Klasse Stift weitere Unterklassen (Figurstift, Planstift) mit entsprechenden Diensten erzeugt und in einem selbst entwickelten Mini-CAD-Programm verwendet.

Im Anschluss daran soll, ausgehend von der Realität, über Objektidentifizierung und Entwurf bis hin zur Implementation, ein größeres Softwareprojekt (Billard) erstellt werden.

Dabei identifizieren die Schülerinnen und Schüler zunächst in einer Entwurfsphase verschiedene Objekte und ordnen sie verschiedenen Klassen zu. Diese Klassen und ihre Beziehungen untereinander werden anschließend in einem Entwurfsdiagramm grafisch dargestellt.

Nach diesem ersten Modellierungsschritt werden über Klassendokumentationen und der Darstellung von Objektkommunikationen anhand von Sequenzdiagrammen Implementationsdiagramme entwickelt. Danach werden die Implementationsdiagramme unter Berücksichtigung der Klassendokumentationen in Javaklassen programmiert.

Abschließend sind die Schülerinnen und Schüler in der Lage, ein eigenes kleines Softwareprojekt (Rotkäppchen) zu entwickeln. Hier soll die Erzeugung eigener Klassen und die Herstellung von Beziehungen weiter eingeübt werden.

Zeitbedarf: 16 67,5-Std. (≙ 24 45-Std.)

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Erzeugung eigener Unterklassen a) Implementierung verschiedener Unterklassen b) Dokumentation der Unterklassen mit Hilfe des Java-Doc-Generators c) Entwicklung eines Programms unter Verwendung Unterklassen	Die Schülerinnen und Schüler <ul style="list-style-type: none"> analysieren und erläutern eine objektorientierte Modellierung (A) ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M) 	<i>Medien/Materialien:</i> Programmiersprache Java, Entwicklungsumgebung BlueJ, UML-Editor <i>Projekte:</i> <ul style="list-style-type: none"> Figurstift Billard Rotkaeppchen
2. Entwicklung eines komplexen grafischen Projekts 2.1 Umsetzung von Anforderungen in Entwurfsdiagramme	<ul style="list-style-type: none"> modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M) 	

<p>a) Aus Anforderungsbeschreibungen werden Objekte mit ihren Eigenschaften identifiziert</p> <p>b) Gleichartige Objekte werden in Klassen (Entwurf) zusammengefasst und um Attribute und Methoden erweitert</p> <p>2.2 Implementationsdiagramme als erster Schritt der Programmierung</p> <p>a) Erweiterung des Entwurfsdiagramms um Konstruktoren und get- und set-Methoden</p> <p>b) Festlegung von Datentypen in Java, sowie von Rückgaben und Parametern</p> <p>c) Entwicklung von Klassendokumentationen mit Hilfe des Java-Doc-Generators</p> <p>d) Erstellung von Sequenzdiagrammen als Vorbereitung für die Programmierung</p> <p>2.3 Implementierung anhand der Dokumentation und des Implementations- u. Sequenzdiagrammes</p> <p>a) Klassen werden in Java-Quellcode umgesetzt</p> <p>b) Das Geheimnisprinzip wird umgesetzt</p> <p>c) Einzelne Klassen und das Gesamtsystem werden anhand der Anforderungen und Dokumentationen auf ihre Korrektheit überprüft.</p> <p>2.4 Vererbungsbeziehungen</p> <p>a) Weitere Vererbungsbeziehungen (Hat-, Kennt-Beziehungen) werden erarbeitet und implementiert</p> <p>b) Ist-Beziehungen werden implementiert (Unterklassen)</p>	<ul style="list-style-type: none"> • modellieren Klassen unter Verwendung von Vererbung (M) • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen (M) • ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M) • stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D) • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I) • interpretieren Fehlermeldungen (D) und korrigieren den Quellcode (I) 	
<p>4. Eigenständige Erarbeitung eines weiteren Softwareprojekts</p>		

Unterrichtsvorhaben EF-IV

Thema: Such- und Sortieralgorithmen auf Feldern 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, auf die in diesem Vorhaben vollständig verzichtet werden soll.

Zunächst lernen die Schülerinnen und Schüler das Feld als eine erste Datensammlung kennen.

Daran anschließend 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 und Vertauschungen auf ihre hin Effizienz untersucht.

Daran anschließend werden die erarbeiteten Strategien systematisiert und im Pseudocode notiert. Die Schülerinnen und Schüler sollen auf diese Weise mindestens 2 der 3 der folgenden Sortierverfahren kennenlernen: das *Sortieren durch Vertauschen*, das *Sortieren durch Auswählen* bzw. das *Sortieren durch Einfügen*.

Des Weiteren soll das Prinzip der *binären Suche* behandelt und nach Effizienzgesichtspunkten untersucht werden.

Zeitbedarf: 10 67,5-Std. (= 15 45-Std.)

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Modellierung von Datensammlungen a) Modellierung von Attributen als Felder b) Deklaration, Instanziierung und Zugriffe auf Felder	Die Schülerinnen und Schüler <ul style="list-style-type: none"> ordnen Attributen lineare Datensammlungen zu (M) analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D) entwerfen Algorithmen zum Suchen und Sortieren (M) beurteilen die Effizienz von Algorithmen hinsichtlich Zeit- und Speicherplatzbedarf (A) 	Datenstruktur: Feld (Array)
2. Explorative Erarbeitung eines Sortierverfahrens a) Sortierprobleme im Kontext informatischer Systeme und im Alltag (z.B. Dateien, Tabellen, Telefonbuch, Adressbuch, usw.) b) Erarbeitung zweier Sortieralgorithmen durch die Schülerinnen und Schüler		Suchverfahren: <ul style="list-style-type: none"> Lineare und Binäre Suche Sortierverfahren: <ul style="list-style-type: none"> Sortieren durch Vertauschen Sortieren durch Auswählen Sortieren durch Einfügen (mindestens zwei der drei Verfahren)

<p>3. Systematisierung von Algorithmen und Effizienzbetrachtungen</p> <p>a) Formulierung verschiedener Algorithmen im Pseudocode</p> <p>b) Effizienzbetrachtungen bzgl. der Rechenzeit und des Speicherplatzbedarfs</p>		
<p>4. Lineare Suche auf unsortierten Daten und Binäre Suche auf sortierten Daten</p> <p>a) Suchaufgaben im Alltag und im Kontext informatischer Systeme</p> <p>b) Effizienzbetrachtungen zur linearen und binären Suche</p>		

Unterrichtsvorhaben EF-VI

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

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

Vorhabenbezogene Konkretisierung:

Zeitbedarf: 6 67,5-Std. (≙ 9 45-Std.)

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Selbständige Erarbeitung von Themen durch die Schülerinnen und Schüler</p> <p>a) Mögliche Themen zur Erarbeitung in Kleingruppen:</p> <ul style="list-style-type: none"> • „Eine kleine Geschichte der Digitalisierung – vom Morzen zum modernen Digitalcomputer“ • „Eine kleine Geschichte der Kryptographie - von Caesar zur Enigma“ • „Kodieren von Texten und Bildern: ASCII, RGB und mehr“ • „Auswirkungen der Digitalisierung – Veränderung der Arbeitswelt und Datenschutz“ <p>b) Vorstellung und Diskussion durch / mit Schülerinnen und Schülern</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (D), • bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A), • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K). 	<p><i>Beispiel:</i> Präsentationen zu informatischen Themen</p> <p>Die Schülerinnen und Schüler bereiten z. B. Bildschirmpräsentationen zu informatischen Themen vor.</p> <p><i>Materialien:</i> Die Schülerinnen und Schüler recherchieren selbstständig im Internet, in der Schülerbibliothek, in öffentlichen Bibliotheken und Museen (z.B. Heinz Nixdorf MuseumsForum), usw.</p>
<p>2. Vertiefung des Themas Datenschutz</p> <p>a) Erarbeitung grundlegender Begriffe des Datenschutzes</p> <p>b) Problematisierung und Anknüpfung an die Lebenswelt der Schülerinnen und Schüler</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

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

Zeitbedarf: 10 67,5-Std. (≅ 15 45-Std.)

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Wiederholung und Erweiterung der objektorientierten Modellierung und Programmierung durch Analyse und Erweiterung eines kontextbezogenen Beispiels</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"> • analysieren und erläutern objektorientierte Modellierungen (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), • ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I) • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen (D), • stellen die Kommunikation zwischen Objekten grafisch dar (D). 	<p><i>Beispiel: Schatzsuche</i></p> <p>Auf dem Bildschirm sind verschiedene Schätze dargestellt als unterschiedliche geometrische Objekte (Kreise, Quadrate, Dreiecke, ...) versteckt, die alle zufällig auf dem Bildschirm angeordnet sind.</p> <p>Wird ein Schatz getroffen (mit der Maus angeklickt), so wird er sichtbar und der Spieler erhält je nach Schatztyp eine gewissen Punktzahl gutgeschrieben. Die Dauer des Spiels kann zeitlich begrenzt werden.</p> <p>Es liegt hier nahe, für die verschiedenen Schätze eine gemeinsame abstrakte Oberklasse zu erzeugen.</p> <p><i>Beispiel: Weihnachtsbaum</i></p> <p>Ein Baum soll mit Kerzen und Kugeln geschmückt werden. Unter dem Baum sollen mehrere Pakete liegen. Wenn eines der Objekte Baum, Kerze, Kugel oder Paket angeklickt wird, soll es reagieren. Der Baum wackelt kurz, die Kugeln ändern ihre Farbe, die Kerzen gehen an oder aus und die Pakete öffnen oder schließen sich.</p> <p>Da sowohl die Kerzen, Kugeln und Pakete als auch der Baum selbst durch unterschiedliche geometrische Objekte dargestellt werden können, die bei Berührung unterschiedlich reagieren, liegt es hier nahe für die verschiedenen Objekte eine gemeinsame abstrakte Oberklasse zu erzeugen.</p>

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:

Listen kommen in Realweltsituationen in allgemeiner und in spezieller Form vor. Sei es eine Warteschlange an einer Kasse, eine Polizeikontrolle oder das Stapeln von Getränkekisten. In allgemeiner Form bei der Zusammenstellung von Zügen oder bei der Aufstellung einer Mannschaft beim Staffellauf. Eine Modellierung derartiger Abläufe stellt daher eine geeignete Möglichkeit dar, Listenstrukturen zu thematisieren.

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

Zeitbedarf: 16 67,5-Std. (= 24 45-Std.)

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse <code>Queue</code></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"> • erläutern Operationen dynamischer (linearer oder nichtlinearer) Datenstrukturen (A), • analysieren und erläutern Algorithmen und Programme (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), • ermitteln bei der Analyse von 	<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>Die Simulationsanwendung stellt eine GUI zur Verfügung, legt ein Wartezimmer an und steuert die Abläufe.</p> <p>Wesentlicher Aspekt des Projek-</p>

	<p>Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</p> <ul style="list-style-type: none"> • modifizieren Algorithmen und Programme (I), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), 	<p>tes ist die Modellierung des Wartezimmers mit Hilfe der Klasse <code>Queue</code>.</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>2. Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse <code>Stack</code></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>	<ul style="list-style-type: none"> • 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 lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D). 	<p><i>Beispiel:</i> Heftstapel</p> <p>In einem Heftstapel soll das Heft einer Schülerin / eines Schülers 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>3. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse <code>List</code></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>		<p><i>Beispiel:</i> Abfahrtslauf</p> <p>Bei einem Abfahrtslauf kommen die Skifahrer nacheinander an und werden nach ihrer Zeit in eine Rangliste eingeordnet. Diese Rangliste wird in einer Anzeige ausgegeben. Ankommende Abfahrer müssen an jeder Stelle der Struktur, nicht nur am Ende oder Anfang eingefügt werden können.</p>

<p>4. Vertiefung - Anwendungen von Listen, Stapeln oder Schlangen in einem weiteren Kontext</p>		<p><i>Beispiel:</i> Skispringen</p> <p>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> <p><i>Beispiel:</i> Terme in Postfix-Notation</p> <p>Die sog. UPN (<i>Umgekehrte-Polnische-Notation</i>) bzw. <i>Post-fix-Notation</i> eines Terms setzt den Operator hinter die Operanden. Um einen Term aus der gewohnten Infixschreibweise in einen Term in UPN umzuwandeln oder um den Wert des Terms zu berechnen, kann ein Stack verwendet werden.</p> <p><i>Beispiel:</i> Rangierbahnhof</p> <p>Auf einem Güterbahnhof gibt es drei Gleise, die nur zu einer Seite offen sind. Waggons können also von einer Seite auf das Gleis fahren und nur rückwärts wieder hinausfahren. Die Waggons tragen Nummern, wobei die Nummer jedoch erst eingesehen werden kann, wenn der Waggon der vorderste an der offenen Gleisseite ist. (Zwischen den Waggons herumzuturnen, um die anderen Waggonnummern zu lesen, wäre zu gefährlich.) Zunächst stehen alle Waggons unsortiert auf einem Gleis. Ziel ist es, alle Waggons in ein anderes Gleis zu fahren, so dass dort die Nummern der Waggons vom Gleisende aus aufsteigend in richtiger Reihenfolge sind. Zusätzlich zu diesen beiden Gleisen gibt es ein Abstellgleis, das zum Rangieren benutzt werden kann.</p>
--	--	---

		<p><i>Beispiel:</i> Autos an einer Ampel zur Zufahrtsregelung</p> <p>Es soll eine Ampel zur Zufahrtsregelung in Java simuliert werden. An einem geradlinigen, senkrecht von unten nach oben verlaufenden Straßenstück, das von Autos nur einspurig in eine Richtung befahren werden kann, ist ein Haltepunkt markiert, an dem die Ampel steht. Bei einem Klick auf eine Schaltfläche mit der Aufschrift „Heranfahren“ soll ein neues Auto an den Haltepunkt heranfahren bzw. bis an das letzte Auto, das vor dem Haltepunkt wartet. Grünphasen der Ampel werden durch einen Klick auf eine Schaltfläche mit der Aufschrift „Weiterfahren“ simuliert. In jeder Grünphase darf jeweils nur ein Auto weiterfahren. Die anderen Autos rücken nach.</p>
--	--	--

Unterrichtsvorhaben Q1-III:

Thema: Suchen und Sortieren auf linearen Datenstrukturen

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

Vorhabenbezogene Konkretisierung:

Die Schülerinnen und Schüler haben im Unterrichtsvorhaben Q1-II die Modellierung und Implementierung linearer Datenstrukturen kennengelernt und angewendet. Außerdem haben sie in der EF einfache Such- und Sortieralgorithmen an realweltlichen Objekten kennengelernt und unter Feldern implementiert. Darauf baut dieses Unterrichtsvorhaben auf.

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

Anschließend werden Sortierverfahren entwickelt und implementiert. Hierbei soll auch ein rekursives Sortierverfahren entwickelt werden. Die Implementationen von Quicksort sowie weiterer elementaren Verfahren wie z.B. dem Sortieren durch Einfügen werden analysiert und erläutert. Falls diese Verfahren in der EF schon entdeckt wurden, sollen sie hier wiedererkannt werden. Die rekursive Abarbeitung eines Methodenaufrufs von Quicksort wird grafisch dargestellt.

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

Zeitbedarf: 14 67,5-Std. (≙ 21 45-Std.)

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Suchen von Daten in Listen</p> <p>a) Lineare Suche in Listen</p> <p>b) Binäre Suche in Listen als Beispiel für rekursives Problemlösen</p> <p>c) Untersuchung der beiden Suchverfahren hinsichtlich ihrer Effizienz (Laufzeitverhalten, Speicherbedarf)</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern Algorithmen und Programme (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M), • modifizieren Algorithmen und Programme (I), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), 	<p><i>Beispiel:</i> Bundesjugendspiele</p> <p>Die Teilnehmer an Bundesjugendspielen nehmen an drei Disziplinen teil und erreichen dort Punktzahlen. Diese werden in einer Wettkampfkarte eingetragen und an das Wettkampfbüro gegeben. Zur Vereinfachung sollte sich das Modell auf die 3 Disziplinen „Lauf“, „Sprung“ und „Wurf“ beschränken. Im Wettkampfbüro wird das Ergebnis erstellt. Das Programm soll dafür zunächst den Besten einer Disziplin heraussuchen können und später das gesamte Ergebnis nach gewissen Kriterien sortieren können.</p> <p>oder</p> <p><i>Beispiel:</i> Karteiverwaltung</p> <p>Für ein Adressverwaltungsprogramm soll eine Methode zum Suchen einer Adresse geschrieben werden.</p>
<p>2. Sortieren in Listen - Entwicklung und Implementierung von iterativen und rekursiven Sortierverfahren</p> <p>a) Implementierung der einfachen Sortierverfahren für eine Liste</p> <p>b) Entwicklung und Implementierung eines rekursiven Sortierverfahren für eine Liste (z.B. Quicksort)</p>	<ul style="list-style-type: none"> • 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), 	<p><i>Beispiel:</i> Bundesjugendspiele (s.o.)</p> <p>oder</p> <p><i>Beispiel:</i> Karteiverwaltung (s.o.)</p>
<p>3. Untersuchung der Effizienz der iterativen und rekursiven Sortierverfahren auf linearen Listen</p> <p>a) Grafische Veranschaulichung der Sortierverfahren</p> <p>b) Untersuchung der Anzahl der Vergleichsoperationen und des Speicherbedarf bei den Sortierverfahren</p> <p>c) Beurteilung der Effizienz der Sortierverfahren</p>	<ul style="list-style-type: none"> • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen (I), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D). 	<p><i>Beispiel:</i> Bundesjugendspiele (s.o.)</p> <p>oder</p> <p><i>Beispiel:</i> Karteiverwaltung (s.o.)</p>

Unterrichtsvorhaben Q1-IV:

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

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

Vorhabenbezogene Konkretisierung:

Bäume bilden eine hierarchische Struktur ab, die man im realen Leben z. B. aus Stammbäumen kennt. Strukturen von Unternehmen sind häufig hierarchisch organisiert (Aufbauorganisation). Weitere Beispiele sind das Inhaltsverzeichnis eines Buches und – aus technischer Sicht – die Struktur des Dateisystems eines Datenträgers. Hierbei handelt es sich um allgemeine Bäume. Ein Sonderfall des allgemeinen Baumes ist der Binärbaum, bei dem jeder Elternknoten höchstens zwei Kindknoten haben kann. Diese Bäume tauchen im realen Leben z. B. dort auf, wo Entscheidungen getroffen werden, die exklusiv mit »ja« oder »nein« beantwortet werden (Entscheidungsbäume). Effizientes Suchen erfolgt im allgemeinen Fall auf Grundlage einer Baumstruktur, so dass in jedem Schritt der Suche die Hälfte der Elemente ausgeschlossen werden kann. Somit ist die Anzahl der Schritte beschränkt mit: $O(\log n)$. Dies trifft auch auf effiziente Sortierverfahren zu, die sich explizit oder implizit den Baum zu nutze machen und in $O(n \log n)$ arbeiten.

Anhand von Beispielen für Baumstrukturen werden zunächst grundlegende Begriffe eingeführt und der rekursive Aufbau binärer Bäume dargestellt.

Anschließend werden für eine Problemstellung in einem der Anwendungskontexte Klassen modelliert und implementiert. Dabei werden die Operationen der Datenstruktur Binärbaum thematisiert und die entsprechende Klasse `BinaryTree` (der Materialien für das Zentralabitur in NRW) verwendet. Klassen und ihre Beziehungen werden in Entwurfs- und Implementationsdiagrammen dargestellt. Die Funktionsweise von Methoden wird anhand grafischer Darstellungen von Binärbäumen erläutert.

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

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

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

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

Zeitbedarf: 16 67,5-Std. (= 24 45-Std.)

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Analyse von Baumstrukturen in verschiedenen Kontexten</p> <p>a) Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit)</p> <p>b) Aufbau und Darstellung von binären Bäumen anhand von Baumstrukturen in verschiedenen Kontexten</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • erläutern Operationen dynamischer (linearer oder nichtlinearer) Datenstrukturen (A), • analysieren und erläutern Algorithmen und Programme (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • modifizieren Algorithmen und Programme (I), 	<p><i>Beispiel:</i> Termbaum</p> <p>Der Aufbau von Termen wird mit Hilfe von binären Baumstrukturen verdeutlicht.</p> <p>oder</p> <p><i>Beispiel:</i> Ahnenbaum</p> <p>Die binäre Baumstruktur ergibt sich daraus, dass jede Person genau einen Vater und eine Mutter hat.</p> <p><i>Weitere Beispiele für Anwendungskontexte für binäre Bäume:</i></p> <p><i>Beispiel:</i> Suchbäume (zur sortierten Speicherung von Daten)</p> <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. (Das gilt für alle Teilbäume.)</p> <p>oder</p> <p><i>Beispiel:</i> Entscheidungsbäume</p> <p>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>oder</p> <p><i>Beispiel:</i> Codierungsbäume für Codierungen, deren Alphabet aus genau zwei Zeichen besteht</p> <p>Morse hat Buchstaben als Folge von Punkten und Strichen codiert.</p>

	<ul style="list-style-type: none"> • 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 lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D). 	<p>Diese Codierungen können in einem Binärbaum dargestellt werden, so dass ein Übergang zum linken Teilbaum einem Punkt und ein Übergang zum rechten Teilbaum einem Strich entspricht. Wenn man im Gesamtbaum startet und durch Übergänge zu linken oder rechten Teilbäumen einen Pfad zum gewünschten Buchstaben sucht, erhält man die Morsecodierung des Buchstabens.</p>
<p>2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse BinaryTree</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>		<p><i>Beispiel: Informatikerbaum als binärer Baum</i></p> <p>In einem <i>binären Baum</i> werden die Namen und die Geburtsdaten von Informatikern lexikographisch geordnet abgespeichert.</p> <p>Alle Namen, die nach dieser Ordnung vor dem Namen im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem Namen im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Das gilt für alle Teilbäume.)</p> <p>Folgende Funktionalitäten werden benötigt:</p> <ul style="list-style-type: none"> • Einfügen der Informatiker-Daten in den Baum • Suchen nach einem Informatiker über den Schlüssel Name • Ausgabe des kompletten Datenbestands in nach Namen sortierter Reihenfolge
<p>3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse BinarySearchTree</p> <p>a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p>		<p><i>Beispiel: Informatikerbaum als Suchbaum</i></p> <p>In einem binären <i>Suchbaum</i> werden die Namen und die Geburtsdaten von Informatikern lexikographisch geordnet abgespeichert.</p>

<p>b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramm, grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften</p> <p>c) Erarbeitung der Klasse <code>BinarySearchTree</code> und Einführung des Interface <code>Item</code> zur Realisierung einer geeigneten Ordnungsrelation</p> <p>d) Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums</p>		<p>Alle Namen, die nach dieser Ordnung vor dem Namen im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem Namen im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Das gilt für alle Teilbäume.)</p> <p>Folgende Funktionalitäten werden benötigt:</p> <ul style="list-style-type: none"> • Einfügen der Informatiker-Daten in den Baum • Suchen nach einem Informatiker über den Schlüssel Name • Ausgabe des kompletten Datenbestands in nach Namen sortierter Reihenfolge
<p>4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen</p>		<p><i>Beispiel:</i> Codierungsbäume (s.o.) oder Huffman-Codierung <i>oder</i> <i>Beispiel:</i> Buchindex</p> <p>Es soll eine Anwendung entwickelt werden, die anhand von Stichworten und zugehörigen Seitenzahlen ein Stichwortregister erstellt. Da die Stichwörter bei der Analyse des Buches häufig gesucht werden müssen, werden sie in der Klasse <code>Buchindex</code> als Suchbaum (Objekt der Klasse <code>BinarySearchTree</code>) verwaltet.</p> <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.) <i>oder</i> <i>Beispiel:</i> Entscheidungsbäume (s.o.) <i>oder</i> <i>Beispiel:</i> Termbaum (s.o.) <i>oder</i> <i>Beispiel:</i> Ahnenbaum (s.o.)</p>

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? Was ist ein Kellerautomat?*

Vorhabenbezogene Konkretisierung:

Endlichen Automaten und formalen Sprachen gehört zum Teilbereich der theoretischen Informatik. Sie bildet einen Grundbaustein für die Berechenbarkeits- und Komplexitätstheorie. Dabei wird untersucht, welche Probleme mit Automaten gelöst werden können.

Die Transduktoren, als Teilgruppe der endlichen Automaten, haben noch einen Bezug zur Lebenswelt den Schülerinnen und Schülern, da mit ihnen zum Beispiel Ticket- oder Getränkeautomaten modelliert werden können. Eine besondere Rolle kommt aber den Akzeptoren zu. Die von ihnen akzeptierten Wörter bilden eine reguläre Sprache, die sich auch über Grammatiken oder reguläre Ausdrücke darstellen lässt. Anwendung finden diese Bereiche unter anderem beim Parsern, die zum Untersuchen von Texten eingesetzt werden.

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 und der Kellerspeicher als Erweiterung des endlichen Automaten eingeführt.

Zeitbedarf: 14 67,5-Std. (≙ 21 45-Std.)

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Endliche Automaten</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>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens auf bestimmte Eingaben (A), analysieren und erläutern Grammatiken regulärer (oder kontextfreier) Sprachen (A), 	<p><i>Beispiele:</i> Cola-Automat, Geldspielautomat,</p> <p>Roboter, Zustandsänderung eines Objekts „Auto“, Akzeptor für bestimmte Zahlen, Akzeptor für Teilwörter in längeren Zeichenketten, Akzeptor für Terme</p>

<p>2. Untersuchung und Entwicklung von Grammatiken regulärer Sprachen</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"> • zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A), • ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A), • entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M), • entwickeln und modifizieren zu einer Problemstellung endliche Automaten (oder Kellerautomaten) (M), • entwickeln zur akzeptierten Sprache eines Automaten die zugehörige Grammatik (M), 	<p><i>Beispiele:</i> reguläre Grammatik für Wörter mit ungerader Parität, Grammatik für Wörter, die bestimmte Zahlen repräsentieren, Satzgliederungsgrammatik</p>
<p>3. Grenzen endlicher Automaten</p> <p>a) Der Kellerautomat (als Erweiterung des endlichen Automaten)</p> <p>b) Entwicklung Kontextfreier Grammatiken</p>	<ul style="list-style-type: none"> • entwickeln zur Grammatik einer regulären (oder kontextfreien) Sprache einen zugehörigen endlichen Automaten (oder Kellerautomaten) (M), • modifizieren Grammatiken regulärer (oder kontextfreier) Sprachen (M), • entwickeln zu einer regulären (oder kontextfreien) Sprache eine Grammatik, die die Sprache erzeugt (M), • stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D), • ermitteln die Sprache, die ein endlicher Automat (oder Kellerautomat) akzeptiert (D). • beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D). 	<p><i>Beispiele:</i> Klammerausdrücke, $a^n b^n$ im Vergleich zu $(ab)^n$</p>

Unterrichtsvorhaben Q2-II:

Thema: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

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

Vorhabenbezogene Konkretisierung:

Datenbanken sind Grundlage vieler Informatiksysteme, mit denen Schülerinnen und Schüler alltäglich in Kontakt kommen. Meist merken sie nichts davon. Soziale Netzwerke, Fahrplanauskunftssysteme, Online-Shops, die Verarbeitung der Patientendaten beim Arzt usw. basieren auf Datenbanksystemen.

Aus diesem Grund erscheint es sinnvoll, anhand eines Realweltproblems in die Thematik einzusteigen. 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: 16 67,5-Std. (= 24 45-Std.)

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Nutzung von relationalen Datenbanken</p> <p>a) Aufbau von Datenbanken und Grundbegriffe</p> <ul style="list-style-type: none"> • Entwicklung von Fragestellungen zur vorhandenen DB • Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, Datenbankschema 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A) • analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A), • analysieren und erläutern eine Datenbankmodellierung (A), 	<p><i>Beispiel:</i> VideoCenter</p> <p>VideoCenter ist die Simulation einer Online-Videothek für den Informatik-Unterricht mit Webfrontends zur Verwaltung der Kunden, der Videos und der Ausleihe. Außerdem ist es möglich direkt SQL-Abfragen einzugeben. Es ist auch möglich, die Datenbank herunter zu laden und lokal zu installieren.</p>

<p>b) SQL-Abfragen</p> <ul style="list-style-type: none"> Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL (<code>SELECT (DISTINCT) ...FROM, WHERE, AND, OR, NOT</code>) auf einer Tabelle Analyse und Erarbeitung von SQL-Abfragen auf einer und mehrerer Tabellen zur Beantwortung der Fragestellungen (<code>JOIN, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, <>, >, <, >=, <=, LIKE, BETWEEN, IN, IS NULL</code>) <p>c) Vertiefung an einem weiteren Datenbankbeispiel</p>	<ul style="list-style-type: none"> erläutern die Eigenschaften normalisierter Datenbankschemata (A), bestimmen Primär- und Sekundärschlüssel (M), ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M), modifizieren eine Datenbankmodellierung (M), modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M), bestimmen Primär- und Sekundärschlüssel (M), überführen Datenbankschemata in vorgegebene Normalformen (M), verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I), ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D), stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D), überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D). 	<p>Unter http://dokumentation.videocenter.schule.de/old/video/index.html findet man den Link zu dem VideoCenter-System sowie nähere Informationen. Lesenswert ist auch die dort verlinkte „Dokumentation der Fallstudie“ mit didaktischem Material, welches alternativ bzw. ergänzend zu der im Folgenden beschriebenen Durchführung verwendet werden kann.</p> <p><i>Beispiel:</i> Schulbuchausleihe</p> <p>Unter www.brd.nrw.de/lerntreffs/informatik/structure/material/sek2/datenbanken.php wird eine Datenbank zur Verfügung gestellt, die Daten einer Schulbuchausleihe enthält (über 1000 Entleiher, 200 Bücher mit mehreren tausend Exemplaren und viele Ausleihvorgänge). Die Datenbank kann in OpenOffice eingebunden werden.</p>
<p>2. Modellierung von relationalen Datenbanken</p> <p>a) Entity-Relationship-Diagramm</p> <ul style="list-style-type: none"> Ermittlung von Entitäten, zugehörigen Attributen, Relationen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms Erläuterung und Modifizierung einer Datenbankmodellierung <p>b) Entwicklung einer Datenbank aus einem Datenbankentwurf</p> <ul style="list-style-type: none"> Modellierung eines relationalen Datenbankschemas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln 		<p><i>Beispiel:</i> Fahrradverleih</p> <p>Der Fahrradverleih <i>BTR (Bikes ToRent)</i> verleiht unterschiedliche Typen von Fahrrädern diverser Firmen an seine Kunden. Die Kunden sind bei <i>BTR</i> registriert (Name, Adresse, Telefon). <i>BTR</i> kennt von den Fahrradfirmen den Namen und die Telefonnummer. Kunden von <i>BTR</i> können CityBikes, Treckingräder und Mountainbikes ausleihen.</p> <p><i>Beispiel:</i> Reederei</p> <p>Die Datenverwaltung einer Reederei soll in einem Datenbanksystem umgesetzt werden. Ausgehend von der Modellierung soll mit Hilfe eines ER-Modells und eines Datenbankschemas dieser erste Entwurf normalisiert und in einem Datenbanksystem umgesetzt werden. Es schließen sich diverse SQL-Abfragen an, wobei auf die Relationenalgebra eingegangen wird.</p>

<p>c) Redundanz, Konsistenz und Normalformen</p> <ul style="list-style-type: none"> • Untersuchung einer Datenbank hinsichtlich Konsistenz und Redundanz in einer Anwendungssituation • Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten) 		<p><i>Beispiel: Buchungssystem</i></p> <p>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.</p> <p>Unter http://mrbs.sourceforge.net findet man ein freies Online-Buchungssystem inklusive Demo, an Hand derer man erläutern kann, worum es in dem Projekt geht.</p> <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 Q2-III:

Thema: Aufbau von und Kommunikation in Netzwerken

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

Vorhabenbezogene Konkretisierung:

Vernetzte Informatiksysteme bestimmen die heutige IT-Infrastruktur und damit haben sie, denkt man zum Beispiel an die zahlreichen Internetanwendungen, auch einen nicht unerheblichen Einfluss auf unsere Lebenswelt.

Ausgehend von alltäglicher Face-to-Face-Kommunikation werden die Grundprinzipien sowie die Bewertungskriterien von Kommunikation erläutert. Das Netzwerk wird als vorteilhafte Kommunikationsstruktur dargestellt und anhand von Topologien und Reichweiten kategorisiert.

Ausgehend davon wird der Protokollbegriff entwickelt und anhand des TCP/IP-Schichtenmodells analysiert. Anschließend wird das Client-Server-Prinzip vorgestellt und angewandt.

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

Zeitbedarf: 8 67,5-Std. (≙ 12 45-Std.)

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Technische Kommunikation als Fortführung natürlicher Kommunikation</p> <p>a) Kommunikation im Sender-Empfänger-Modell</p> <p>b) Kriterien von technischen Kommunikationsarten</p> <p>c) Die Geschichte der technischen Kommunikation</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> - beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A), - nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D), 	<p><i>Projekteinstieg:</i></p> <p>Kommunikation im Wilden Westen</p> <p>Shannon-Weaver Modell</p>
<p>2. Aufbau von Netzwerken und Kommunikationsregeln</p> <p>a) Das Netzwerk als Organisationsprinzip der Kommunikation und Möglichkeiten der Ausformung</p> <p>b) Geregelte technische Kommunikation durch Protokolle in Schichtenmodellen</p>	<ul style="list-style-type: none"> - analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A). 	<p><i>Beispiele zu Netzwerk-Topologien:</i></p> <ul style="list-style-type: none"> • Bus-Topologie • Ring-Topologie • Stern-Topologie • Baum-Topologie • Maschen-Topologie
<p>3. Aufgabenteilung in Netzwerken durch Server und Client</p> <p>a) Aufbau und Aufgaben der Client-Server-Struktur</p> <p>b) Protokolle zwischen Client und Server</p>		<p><i>Beispiele zu Referenzmodellen für Netzwerkprotokolle als Schichtenarchitektur: OSI, TCP/IP</i></p> <p><i>Beispiele für Client-Server-Protokolle: NFS, RFS, APF, SMB</i></p>
<p>4. Kryptologie</p> <p>a) Veranschaulichen und Anwenden von symmetrischen und asymmetrischen kryptographischen Verfahren</p> <p>b) Bewertung der Verfahren hinsichtlich ihrer Sicherheit und ihrem Aufwand</p>		<p><i>Beispiele für Verschlüsselungsverfahren</i></p> <ul style="list-style-type: none"> • Caesar-Chiffre • Vigenère-Verfahren • RSA-Verschlüsselung • mit PGP/GPG verschlüsselte Daten

Unterrichtsvorhaben Q2-IV:

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

2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

Die Fachkonferenz Informatik des St.-Franziskus-Gymnasiums hat die folgenden fachmethodischen und fachdidaktischen Grundsätze beschlossen. In diesem Zusammenhang beziehen sich die Grundsätze 1 bis 12 auf fächerübergreifende Aspekte, die auch Gegenstand der Qualitätsanalyse sind, die Grundsätze 13 bis 28 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) Die Lehr- und Lernzeit wird intensiv für Unterrichtszwecke genutzt.
- 12) Es herrscht ein positives Lernklima im Unterricht.

Fachliche Grundsätze:

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

2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

2.3.1 Grundsätze der Leistungsbewertung

Auf der Grundlage von §13 - §16 der APO-GOSt sowie Kapitel 3 des Kernlehrplans Informatik für die gymnasiale Oberstufe hat die Fachkonferenz des St.-Franziskus-Gymnasiums im Einklang mit den allgemeinen Grundlagen der Leistungsbemessung und -bewertung am St.-Franziskus-Gymnasium die nachfolgenden Grundsätze zur Leistungsbewertung beschlossen.

- Lernerfolgsüberprüfungen sind ein kontinuierlicher Prozess. Bewertet werden alle im Zusammenhang mit dem Unterricht erbrachten Leistungen (schriftliche Arbeiten, mündliche Beiträge, praktische Leistungen).
- Leistungsbewertung bezieht sich auf die im Unterricht geförderten Kompetenzen.
- Die Lehrperson gibt den Schülerinnen und Schülern im Unterricht hinreichend Gelegenheit, die entsprechenden Anforderungen der Leistungsbewertung im Unterricht in Umfang und Anspruch kennenzulernen und sich auf sie vorzubereiten.
- Bewertet werden der Umfang, die selbstständige und richtige Anwendung der Kenntnisse, Fähigkeiten und Fertigkeiten sowie die Art der Darstellung.

2.3.2 Formen der Leistungsüberprüfung

2.3.2.1 Beurteilungsbereich Klausuren

Klausuren dienen der schriftlichen Überprüfung der Lernergebnisse einer vorausgegangenen Unterrichtssequenz. Sie sind so anzulegen, dass Sachkenntnisse und methodische Fertigkeiten nachgewiesen werden können. Sie bedürfen einer angemessenen Vorbereitung und verlangen klare Aufgabenstellungen. Im Umfang und Anforderungsniveau sind Klausuren abhängig von den kontinuierlich ansteigenden Anforderungen entsprechend dem Lehrplan.

Es ist darauf zu achten, dass nicht nur die Richtigkeit der Ergebnisse und die inhaltliche Qualität, sondern auch die angemessene Form der Darstellung unabdingbare Kriterien der Bewertung der geforderten Leistung sind.

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

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

Am St.-Franziskus-Gymnasium werden die Kursarbeiten bzw. Klausuren in der Regel nach einem vorab festgelegten Punkteschema bewertet. Dabei wird eine glatt ausreichende Leistung bei 45% der Punktzahl erreicht. Die übrigen Notenstufen ergeben sich dann dadurch, dass für jede Notenstufe Intervalle der erreichten Punkte gebildet werden, die in der Regel gleich groß sind (vgl. die linke Tabelle auf der nachfolgenden Seite).

Spätestens in der Abiturvorklausur wird die im Zentralabitur vorgegebene Zuordnung der erreichten Punkte (maximale Punktzahl: 100 im GK) zur Note gemäß der nachfolgend aufgeführten rechten Tabelle als Grundlage der Notenfindung genutzt.

≥	Punkte
0 %	0
20 %	1
26,7 %	2
33,3 %	3
40 %	4
45 %	5
50 %	6
55 %	7
60 %	8
65 %	9
70 %	10
75 %	11
80 %	12
85 %	13
90 %	14
95 %	15

Punkte	Note
0-19	6
20-26	5-
27-33	5
34-39	5+
40-44	4-
45-49	4
50-54	4+
55-59	3-
60-64	3
65-69	3+
70-74	2-
75-79	2
80-84	2+
85-89	1-
90-94	1
95-100	1+

Die Fachkonferenz Informatik legt die Dauer der Kursarbeiten und Klausuren fest. Am St.-Franziskus-Gymnasium gelten für die Sekundarstufe II folgende Regelungen:

Jahrgangsstufe	1. Klausur, 1. HJ	2. Klausur, 1. HJ	1. Klausur, 2. HJ	2. Klausur, 2. HJ
EF	---	90 Minuten	90 Minuten	90 Minuten
Q1 GK	90 Minuten	90 Minuten	90 Minuten	90 Minuten
Q2 GK	135 Minuten	135 Minuten	135 Minuten	---

Gemäß dem Beschluss der Lehrerkonferenz kann in der Qualifikationsphase I die erste Klausur im 2. Halbjahr durch eine Facharbeit ersetzt werden.

2.3.2.2 Beurteilungsbereich Sonstige Mitarbeit

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

Leistungsaspekte

Mündliche Leistungen

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

Praktische Leistungen am Computer

- Implementierung, Test und Anwendung von Informatiksystemen

Sonstige schriftliche Leistungen

- Lernerfolgsüberprüfung durch kurze schriftliche Übungen
Schriftliche Übung dauern ca. 20 Minuten und umfassen den Stoff der letzten ca. 4 – 6 Stunden.

Kriterien

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

Die Bewertungskriterien stützen sich auf

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

Besonderes Augenmerk ist dabei auf

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

Bei Gruppenarbeiten auch auf

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

Bei Projektarbeit darüber hinaus auf

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

2.3.3 Grundsätze der Leistungsrückmeldung und Beratung

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

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

Die Leistungsrückmeldung kann

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

erfolgen.

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

2.3.4 Individuelle Förderung

Die Lehrer beobachten die individuellen Leistungen in allen Bereichen der Informatik über einen längeren Zeitraum, um auf dieser Grundlage ein Leistungsbild zu erhalten. Neben der Orientierung an den Kompetenzstandards der jeweiligen Jahrgangsstufe kann bei der Leistungsbewertung auch die jeweilige Entwicklung des Schülers bzw. der Schülerin gemäß der zu beobachtenden Lern- und Denkfortschritte berücksichtigt werden.

Der Informatikunterricht lebt von der verantwortungsvollen und selbstständigen Arbeit der Schülerinnen und Schüler, so dass die Lehrperson die nötige Zeit hat, bei Bedarf gezielt und individuell zu fördern.

Leistungsstärkere Schülerinnen und Schüler können ihr Wissen anhand von vertiefenden Problemstellungen erweitern.

2.3.5 Bildung der Zeugnisnote

In die Endnote gehen alle im Unterricht erbrachten Leistungen ein. Dabei nehmen die Beurteilungen der Klausuren den gleichen Stellenwert ein wie die Leistungen im Bereich der Mitarbeit im Unterricht. Zudem ist bei der Notenfindung die individuelle Lernentwicklung der Schülerinnen und Schüler angemessen zu berücksichtigen.

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 Zusammenarbeit mit den Fächern Sozialwissenschaften und Philosophie ausgelotet werden.

Vorbereitung auf die Erstellung der Facharbeit

Möglichst schon im zweiten Halbjahr der Einführungsphase, spätestens jedoch im ersten Halbjahr des ersten Jahres der Qualifikationsphase werden im Unterricht an geeigneten Stellen Hinweise zur Erstellung von Facharbeiten gegeben. Das betrifft u. a. Themenvorschläge, Hinweise zu den Anforderungen und zur Bewertung. Es wird vereinbart, dass nur Facharbeiten vergeben werden, die mit der eigenständigen Entwicklung eines Softwareproduktes verbunden sind.

Exkursionen

In der Einführungsphase wird im Rahmen des Unterrichtsvorhabens „Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes“ eine Exkursion zum Heinz Nixdorf MuseumsForum durchgeführt. Die außerunterrichtliche Veranstaltung wird im Unterricht vor- und nachbereitet.

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 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.

Nach Abschluss des Abiturs 2017 hat die Fachkonferenz Informatik im November 2017 auf der Grundlage ihrer Unterrichtserfahrungen eine Gesamtsicht des schulinternen Curriculums vorgenommen. Dabei wurde festgestellt, dass bei den bisherigen Durchgängen das Thema Q2-III "Aufbau von und Kommunikation in Netzwerken", welches zur Zeit nicht abiturrelevant ist, aus zeitlichen Gründen nicht behandelt werden konnte. Es wurden aber zunächst keine Kürzungen oder Änderungen in der Themenauswahl vorgenommen, sodass das vorliegende schulinterne Curriculum bis zur nächsten Evaluation als verbindlich angesehen werden kann.

Für die Fachschaft Informatik

im November 2017 Dietmar Müller (Fachvorsitzender)