

Schulinterner Lehrplan zum Kernlehrplan für die gymnasiale Oberstufe

Informatik

Inhaltsverzeichnis

1 Die Fachgruppe Informatik des LvD-Gymnasiums informiert:	2
2 Entscheidungen zum Unterricht	3
<i>2.1) Unterrichtsvorhaben und Kooperationsvereinbarungen</i>	<i>3</i>
<i>2.1.1 Übersichtsraster Unterrichtsvorhaben</i>	<i>4</i>
<i>2.1.2 Konkretisierte Unterrichtsvorhaben</i>	<i>12</i>
2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit.....	60
2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung.....	61
<i>2.3.2 Beurteilungsbereich Sonstige Mitarbeit</i>	<i>62</i>
3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen	66
3.1 Zusammenarbeit mit anderen Fächern.....	66
3.2 Vorbereitung auf die Erstellung der Facharbeit.....	66
3.3 Exkursionen und MINT-Tage.....	66

1 Die Fachgruppe Informatik des LvD-Gymnasiums informiert:

Das Luise-von-Duesberg-Gymnasium (LvD) ist als dreizügige Schule in Kempen (am linken Niederrhein) angelegt. Knapp 800 Schülerinnen und Schülern werden von knapp 100 Lehrerinnen und Lehrern unterrichtet. Das Einzugsgebiet der Schule umfasst auch die Nachbargemeinden. Im Bereich der Sekundarstufe II kooperiert das LvD mit seiner Nachbarschule dem Thomaeum und bietet mit ihm zahlreiche gemeinsame Kurse an, unter anderem in Informatik.

Das Fach Informatik wird am LvD ab Klasse 5 unterrichtet. In der Jahrgangsstufe 8 und 9 findet der Unterricht im Wahlpflichtbereich II zweistündig (Stufe 8) bzw. dreistündig (Stufe 9) statt.

In der Sekundarstufe II bietet das LvD Grundkurse und Leistungskurse in Informatik an.

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

Durch projektartiges Vorgehen, offene Aufgaben und Möglichkeiten, Problemlösungen zu verfeinern oder zu optimieren, entspricht der Informatikunterricht 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 LvD aus sieben Lehrkräften. In zwei Computerräumen mit 16 bzw. 14 Schülerarbeitsplätzen und einigen Nebenräumen (Selbstlernzentrum, Physikraum usw.) kann der Fachunterricht stattfinden. Die Arbeitsplätze in den beiden Fachräumen sind an ein schulinternes Netzwerk angeschlossen, das den Schülerinnen und Schülern sowohl einen individuellen, kennwortgeschützten Speicherbereich zur Verfügung stellt, als auch einen nur-lesbaren Bereich mit den momentan benötigten Materialien und den aktuellen digitalen Tafelbildern. Durch die Öffnung des Servers nach außen ist ein verschlüsselter Zugang zu den Speicherbereichen jederzeit auch übers Internet möglich.

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

2 Entscheidungen zum Unterricht

2.1) Unterrichtsvorhaben und Kooperationsvereinbarungen

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

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

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

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

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

Infolge der Kooperation der benachbarten Gymnasien Thomaeum und LvD in der Q1 und Q2 sind die folgenden Unterrichtsvorhaben der Einführungsphase in gemeinsamen Fachkonferenzen entwickelt worden und unterliegen einem ständigen Abgleich mit den aktuellen Gegebenheiten der beiden Schulen.

Für die Qualifikationsphase wurden hinsichtlich der Kooperation und zur Erleichterung für wiederholende Schülerinnen und Schülern die folgende Reihenfolge der Unterrichtsvorhaben für beide Schulen verbindlich festgelegt.

Qualifikationsphase 1.1: Vorhaben Q1-I, Q1-II, Q1-III-Beginn

Qualifikationsphase 1.2: Vorhaben Q1-III-Ende, Q1-IV, Q1-V

Qualifikationsphase 2.1: Vorhaben Q2-I, Q2-II, Q2-III-Beginn

Qualifikationsphase 2.2: Vorhaben Q2-III, Q2-IV

2.1.1 Übersichtsraster Unterrichtsvorhaben

I) Einführungsphase

Einführungsphase	
<p><u>Unterrichtsvorhaben E-I</u></p> <p>Thema: <i>Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten</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• Dateisystem• Internet• Einsatz von Informatiksystemen <p>Zeitbedarf: 10 Stunden</p>	<p><u>Unterrichtsvorhaben E-II</u></p> <p>Thema: <i>Grundlagen der objektorientierten Analyse und Modellierung</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none">• Modellieren• Implementieren• Darstellen und Interpretieren• Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none">• Daten und ihre Strukturierung• Darstellen und Interpretieren <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none">• Objekte und Klassen• Beziehungen von Objekten und von Klassen• Modellierung unter Beachtung grafischer Darstellungsformen (Diagrammen) <p>Zeitbedarf: 12 Stunden</p>

Einführungsphase

Unterrichtsvorhaben E-III

Thema:

Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen in Java

Zentrale Kompetenzen:

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

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

- Objekte, Klassen und ihre grundlegenden Beziehungen (Handlungspfeile bzw. einfache gerichtete Assoziationen mit Kardinalität)
- Implementation der Klassenmethoden inklusive Parametrisierung und Sichtbarkeiten (private/public)
- Syntax und Semantik einer Programmiersprache
- Prinzip und Implementation von parametrisierten Konstruktoren
- Set-/get-Methoden unter Beachtung der Sichtbarkeiten (public/private) für den Zugriff auf Attribute

Zeitbedarf: 28 Stunden

Unterrichtsvorhaben E-IV

Thema:

Such- und Sortieralgorithmen anhand kontextbezogener Beispiele

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Algorithmen und deren Umsetzung

Inhaltliche Schwerpunkte:

- Strategien und Algorithmen zum Suchen und Sortieren auf Feldern
- Analyse, Entwurf und Implementierung einfacher Algorithmen

Zeitbedarf: 9 Stunden

Einführungsphase

Unterrichtsvorhaben E-V

Thema:

Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes

Zentrale Kompetenzen:

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatik, Mensch und Gesellschaft
- Informatiksysteme

Inhaltliche Schwerpunkte:

- Wirkungen der Automatisierung
- Geschichte der automatischen Datenverarbeitung
- Digitalisierung

Zeitbedarf: 15 Stunden

Summe Einführungsphase: 74

II) Qualifikationsphase (Q1 und Q2) – GRUNDKURS

Qualifikationsphase 1	
<p><u>Unterrichtsvorhaben Q1-I</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 • 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: 20 Stunden</p>	<p><u>Unterrichtsvorhaben Q1-II</u></p> <p>Thema: <i>Vertiefung 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 • Formale Sprachen und Automaten • 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: 8 Stunden</p>

Qualifikationsphase 1

Unterrichtsvorhaben Q1-III

Thema:

Suchen und Sortieren auf linearen Datenstrukturen

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Algorithmen
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

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

Zeitbedarf: 16 Stunden

Unterrichtsvorhaben Q1-IV

Thema:

Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

Zentrale Kompetenzen:

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

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Zeitbedarf: 20 Stunden

Qualifikationsphase 1

Unterrichtsvorhaben Q1-V

Thema:

Sicherheit und Datenschutz in Netzstrukturen

Zentrale Kompetenzen:

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

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

Zeitbedarf: 10 Stunden

Summe Qualifikationsphase 1: 74 Stunden

Qualifikationsphase 2

Unterrichtsvorhaben Q2-I

Thema:

Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen

Zentrale Kompetenzen:

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

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Zeitbedarf: 24 Stunden

Unterrichtsvorhaben Q2-II

Thema:

Endliche Automaten und formale Sprachen

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Endliche Automaten und formale Sprachen

Inhaltliche Schwerpunkte:

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

Zeitbedarf: 16 Stunden

Qualifikationsphase 2

Unterrichtsvorhaben Q2-III

Thema:

Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

Zentrale Kompetenzen:

- Argumentieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Einzelrechner und Rechnernetzwerke
- Grenzen der Automatisierung

Zeitbedarf: 12 Stunden

Unterrichtsvorhaben Q2-IV

Thema:

Wiederholungsblock unter Beachtung der aktuellen Zentralabiturvorgaben

Zentrale Kompetenzen:

- Argumentieren
- Kommunizieren und Kooperieren
- Implementieren

Zeitbedarf: 12 Stunden

Summe Qualifikationsphase 2: 64 Stunden

2.1.2 Konkretisierte Unterrichtsvorhaben

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

Verbindliche Festlegungen der Fachkonferenz: Die Fachkonferenz hat Themen, Leitfragen und die Ausführungen unter der Überschrift *Vorhabenbezogene Konkretisierung* verbindlich vereinbart, ebenso die Sequenzierung der Unterrichtsvorhaben (erste Tabellenspalte) und die ausgewiesenen Kompetenzen (zweite Tabellenspalte). Alle Mitglieder der Fachkonferenz haben sich darauf verständigt, in ihrem Unterricht Lerngelegenheiten anzubieten, so dass Schülerinnen und Schüler diese Kompetenzen im Rahmen der festgelegten Unterrichtssequenzen erwerben oder vertiefen können.

Unterrichtliche Anregungen: Die angeführten Beispiele, Medien und Materialien sind dagegen Vorschläge bzw. Hilfen für die Lehrkräfte. In diesen Bereichen sind Abweichungen von den vorgeschlagenen Vorgehensweisen möglich.

In der Einführungsphase und Qualifikationsphase werden unterschiedliche didaktische Lernumgebungen verwendet. Exemplarisch seien an dieser Stelle die Software „BlueJ“, „Fujaba Life3“, „Greenfoot“ und „Filius“ genannt. Die benötigten Installationspakete und Dokumentationen werden den Schülerinnen und Schülern für die Arbeit in der Schule und zuhause von der jeweiligen Lehrkraft zur Verfügung gestellt.

In der Qualifikationsphase werden bei der Umsetzung der Unterrichtsvorhaben Anpassungen gemäß der aktuellen Vorgaben für das Zentralabitur Informatik in NRW vorgenommen und sollten bei der folgenden Konkretisierung beachtet werden.

I) Einführungsphase

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

Die Schülerinnen und Schüler

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

Unterrichtsvorhaben EF-I

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

Leitfragen: *Wie werden Daten in informatischen Systemen abgebildet und bearbeitet? Wie kann die in der Schule vorhandene informatische Ausstattung genutzt werden?*

Vorhabenbezogene Konkretisierung:

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

Den Rahmen zur Betrachtung von Daten und deren Bearbeitung mittels Algorithmen stellt ein Projekt zum Themengebiet "Such- und Sortierverfahren" dar.

Zunächst wird auf den Begriff der Datenspeicherung eingegangen und auf die Probleme bei der automatisierten Suche bzw. der Sortierung von Daten.

Im Vergleich zu menschlichen Ansätzen werden Standardalgorithmen von den Schülerinnen und Schülern referiert und für eine digitale Präsentation aufbereitet. Dabei lernen die Schülerinnen und Schüler die gezielte Internetrecherche, digitale Präsentationstechniken und unterschiedliche Vortragsformen im Kontext der Kommunikation und Kooperation mit ihren Gruppenmitgliedern kennen und nutzen.

Ein Vergleich der referierten Verfahren führt die Schülerinnen und Schüler zu wesentlichen Aspekten der Effizienzbewertung von Algorithmen.

Zeitbedarf: 10 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Information, deren Kodierung und Speicherung</p> <p>(a) Informatik als Wissenschaft der Verarbeitung von Informationen</p> <p>(b) Darstellung von Informationen in Schrift, Bild und Ton</p> <p>(c) Speichern von Daten mit informatischen Systemen am Beispiel der Schulrechner</p> <p>(d) Vereinbarung von Richtlinien zur Datenspeicherung auf den Schulrechnern (z.B. Ordnerstruktur, Dateibezeichner usw.)</p>	<p>Die Schülerinnen und Schüler</p> <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) • kommunizieren, kooperieren und strukturieren bei der Präsentation und Darstellung der Algorithmen 	<p><i>Beispiel:</i> Speicherung von Daten zum automatisierten Suchen und Sortieren</p>
<p>2. Informations- und Datenübermittlung in Netzen</p> <p>(a) Projekt zu ausgewählten Standardalgorithmen (z.B. Such- und Sortierverfahren), die recherchiert und präsentiert werden.</p> <p>(b) Informatische Kommunikation in Rechnernetzen am Beispiel des Schulnetzwerks (z.B. Benutzeranmeldung, Netzwerkordner, Zugriffsrechte, Client-Server)</p> <p>(c) Richtlinien zum verantwortungsvollen Umgang mit dem Internet</p>		<p><i>Beispiel:</i> Nutzung von gegebenen Standardalgorithmen und deren Aufbereitung zur Präsentation vor einer Gruppe</p>

3. Aufbau informatischer Systeme

- (a) Identifikation des EVA-Prinzips (Eingabe-Verarbeitung-Ausgabe) als Prinzip der Verarbeitung von Daten.

Material: Demonstrationshardware

Durch die Analyse und Demontage eines Demonstrationsrechners entdecken die Schülerinnen und Schüler verschiedene Hardwarekomponenten. Als Demonstrationsrechner bietet sich ein ausrangierter Schulrechner an.

Unterrichtsvorhaben EF-II

Thema: Grundlagen der objektorientierten Analyse und Modellierung

Leitfrage: *Wie lassen sich Gegenstandsbereiche informatisch modellieren und im Sinne der Objektorientierung darstellen?*

Vorhabenbezogene Konkretisierung:

Ein zentraler Bestandteil des Informatikunterrichts der Einführungsphase ist die Objektorientierte Modellierung. Dieses Unterrichtsvorhaben führt in die Grundlagen der Analyse und Modellierung von Objekten und Klassen in diesem Kontext ein.

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

Dies erfolgt im Rahmen eines Projektes mit Hilfe einer didaktischen Lernumgebung. Dazu muss der grundlegende Aufbau von Objekten thematisiert und deren Klassifikation mittels Deklaration, Initialisierung und Methodenaufrufen unterschieden werden.

Da bei der Umsetzung dieses ersten Unterrichtsvorhabens zur OOM konsequent auf die Aspekte der Implementation verzichtet wird und grafische Darstellungsformen verwendet werden, ist auf diese Weise eine Fokussierung auf die Grundlagen der Objektorientierung möglich, ohne dass syntaktische Feinheiten ablenken.

Zeitbedarf: 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Identifikation von Objekten</p> <p>(a) Unter Verwendung eines lebensweltnahen Beispiels werden Objekte und deren Beziehungen im Sinne der Objektorientierten Modellierung eingeführt.</p> <p>(b) Objekte werden visualisiert und mit sinnvollen Attributen versehen</p> <p>(c) Objektfähigkeiten werden als Objektbeziehung (die späteren Methoden) dargestellt.</p> <p>(d) Prinzipiell typgleiche Objekte werden analysiert und deren Zusammenfassung zu Klassen untersucht.</p> <p>(e) Der Zusammenhang zwischen Objektdiagrammen und den Klassen eines Klassendiagrammes bezüglich Name, Attribute und Methoden wird schematisiert.</p> <p>(f) Vertiefung: Modellierung weiterer Beispiele ähnlichen Musters</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), • stellen die Kommunikation zwischen Objekten grafisch dar (D), • stellen den Zustand eines Objekts dar (D). 	<p><i>Beispiel:</i></p> <p><i>Strategiespiel in der Art der Schlag-den-Raab-Spiele</i></p> <p>Schülerinnen und Schüler entwickeln ein eigenes Strategiespiel, bei dem gewürfelt wird, bis die Summe 42 überschritten ist oder der Spieler aufhört. Bei mehreren Spielern gewinnt derjenige, dessen Summe kleiner als 42 ist und näher an der 42 liegt.</p> <p>Schülerinnen und Schüler entwickeln Spiele ohne graphische Oberfläche als Menge von Objekten mit resultierenden Objektmanipulationen, die zu Klassen mit Attributen und Methoden zusammen gefasst werden können.</p> <p><i>Materialien:</i></p> <p><i>Unterrichtsskripte zur Entwicklung eines objektorientierten Softwareprojektes</i></p>

<p>2. Erweiterung des Klassendiagrammes um Klassenbeziehungen</p> <p>(g) Unter Verwendung eines lebensweltnahen Beispiels werden verschiedene Klassenbeziehungen als Rechtesystem für den Zugriff von Objekten auf andere Objekte im Sinne der Objektorientierten Modellierung eingeführt.</p> <p>(h) Klassenbeziehungen werden visualisiert und interpretiert.</p> <p>(i) Vertiefung: Modellierung weiterer Beispiele ähnlichen Musters</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • ermitteln bei der Analyse einfacher Problemstellungen Klassenbeziehungen (M), • modellieren Klassenbeziehungen mit ihren Namen, Richtungen und Kardinalitäten (M), • stellen die Zugriffsrechte zwischen Objekten von Klassen grafisch dar (D). 	<p><i>Materialien:</i></p> <p><i>BlueJ, Fujaba Life3 oder ähnliche Umgebungen mit zugehörigen Dokumentationen und Unterrichtsskripten</i></p> <p><i>Beispiel:</i></p> <p><i>Strategiespiel in der Art der Schlag-den-Raab-Spiele</i></p> <p>Schülerinnen und Schüler analysieren, welche Objekte welche Art von Zugriff auf andere Objekte im Rahmen ihres Projektes benötigen und veranschaulichen diese grafisch. Sie erkennen die Notwendigkeit einer Masterklasse für ihr Projekt, welche die Verwaltung des Spiels und aller Objekte übernimmt.</p> <p>Schülerinnen und Schüler analysieren ihr Projekt auf Sicherheitslücken, die sich z.B. durch Kreise bei Klassenbeziehungen ergeben.</p>
---	---	--

Unterrichtsvorhaben EF-III

Thema: Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen in Java

Leitfragen: *Wie lassen sich Objekte auf der Basis von Klassen inklusive ihrer Beziehungen realisieren und wie können die Methoden algorithmisch umgesetzt werden?*

Vorhabenbezogene Konkretisierung:

Der Schwerpunkt dieses Unterrichtsvorhabens liegt in der Implementierung der modellierten Klassen inklusive ihres Zugriffssystems sowie der Objektbeziehungen als Klassenmethoden.

Mit Hilfe einer didaktischen Lernumgebung, welche die Instanziierung von Objekten und Klassenbeziehungen sowie den Aufruf von Methoden durch einen Objektinspektor erlaubt, setzen die Schülerinnen und Schüler ihr Strategiespiel um.

Die Darstellung der Methoden erfolgt je nach verwendeter Lernumgebung in Java oder in einer quelltextfreien algorithmischen Form.

Wesentlich sind die Nutzung von Kontrollstrukturen in Form von Schleifen und Verzweigungen zur Steuerung des Methodenablaufes sowie die Nutzung von Objektmanipulationsarten. Speziell lernen die Schülerinnen und Schüler die Erzeugung und Zerstörung von Objekten und Klassenbeziehungen sowie die Identifikation von Objekten im Speicher entlang von Ketten aus den initialisierten gerichteten Klassenbeziehungen. In diesem Zusammenhang wird auch auf die Identifikation von Objekten anhand ihrer Attributwerte eingegangen.

Das Unterrichtsvorhaben schließt mit der Vorstellung und Besprechung der Projektergebnisse und der Projektmappen.

Zeitbedarf: 28 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Umsetzung der Klassen in der Entwicklungsumgebung</p> <p>(a) Umsetzung einer Klasse mit Attributen und Methoden</p> <p>(b) Unterscheidung von einfachen Methoden und Konstruktoren</p> <p>(c) Erzeugung von Objekten und Links (als dynamisch erzeugte Klassenbeziehungen)</p> <p>(d) Thematisierung des Geheimnisprinzips und des Autonomitätsprinzips von Objekten</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), • ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), 	<p><i>Beispiel:</i></p> <p>Strategiespiel in der Art der Schlag-den-Raab-Spiele</p> <p>Schülerinnen und Schüler setzen ein Spiel um, bei dem man durch Würfeln die aktuelle Gesamtsumme möglichst nah an die 42 bringen muss. Der Gegner ist ein anderer Spieler oder der Computer.</p> <p>Schülerinnen und Schüler erstellen eine Projektmappe bestehend aus der Spielbeschreibung (Ziel, Bedingungen und Ablauf), dem benötigten Klassendiagramm, Methodenspezifikationen, Algorithmen, Beschreibung des Programmablaufs und der Stundenprotokolle.</p>
<p>2. Algorithmische Umsetzung der Methoden</p> <p>(a) Erzeugung und Manipulation von Objekten</p> <p>(b) Nutzung von Schleifen und Alternativen zur bedingten Erzeugung von Objekten</p> <p>(c) Nutzung von Parametern zur Beeinflussung des Programmverhalten</p>	<ul style="list-style-type: none"> • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M), • ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M), • stellen Klassen und Klassenbeziehungen in Diagrammen grafisch dar (D), 	<p>Sie stellen ihr Projekt durch Vorführung des Endproduktes vor, inklusive Erläuterung der Diagramme und Methoden.</p>
<p>3. Präsentation der Projektergebnisse</p> <p>(a) exemplarische Vorführung des gestalteten Spiels</p> <p>(b) Vorstellung des zu Grunde liegenden Klassendiagrammes</p>	<ul style="list-style-type: none"> • modifizieren einfache Algorithmen und Programme (I), • verwenden Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen im Rahmen der Methodenumsetzungen (I), • setzen einfache Algorithmen unter Beachtung der 	

<p>(c) (exemplarische) Darstellung der wesentlichen Methoden</p> <p>(d) Besprechung der Projektdokumentationen / Projektmappen</p>	<p>Syntax und Semantik einer Programmiersprache in der Lernumgebung um (I),</p> <ul style="list-style-type: none">• testen Programme schrittweise anhand von Beispielen (I),• interpretieren Fehlermeldungen und korrigieren ihre Lösungen (I),• dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D).	
--	--	--

Unterrichtsvorhaben EF-IV

Thema: Such- und Sortieralgorithmen anhand kontextbezogener Beispiele

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

Vorhabenbezogene Konkretisierung:

Dieses Unterrichtsvorhaben beschäftigt sich mit der Implementation von Such- und Sortieralgorithmen. Der Schwerpunkt des Vorhabens liegt dabei auf den einfachen elementaren Verfahren, wobei zusätzliche effiziente Verfahren zur Abgrenzung optional sind.

Im Rahmen der Betrachtung möglicher Einsatzszenarien für Such- und Sortieralgorithmen betrachten die Schülerinnen und Schüler die elementaren Such- und Sortierverfahren der quadratischen Aufwandsklasse und bestimmen deren Effizienz unter Beachtung der Anzahl notwendiger Vergleiche sowie des benötigten Speicherplatzes. Auf die Betrachtung konkreter Aufwandsklassen sowie des O-Kalküls wird verzichtet.

Daran anschließend werden die Verfahren in der gewählten Programmiersprache umgesetzt. Dadurch lernen die Schülerinnen und Schüler den Aufbau und Syntax der Sprache kennen und anzuwenden.

Die Reihenfolge der Sequenzblöcke wird explizit nicht festgelegt, so dass beispielsweise die Reihenfolge der Such- und Sortierverfahren beliebig ist.

Zeitbedarf: 9 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Explorative Erarbeitung eines Sortierverfahrens</p> <p>(a) Sortierprobleme im Kontext informatischer Systeme und im Alltag (z.B. Dateisortierung, Tabellenkalkulation, Telefonbuch, Bundesligatabelle, usw.)</p> <p>(b) Vergleich zweier Elemente als Grundlage eines Sortieralgorithmus</p> <p>(c) Betrachtung verschiedener elementarer Sortieralgorithmen</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeit und Speicherplatzbedarf (A), • entwerfen Algorithmen zum Suchen bzw. Sortieren (M), • analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D), • nutzen die im Unterricht eingesetzten Informatiksysteme selbständig, sicher, zielführend und verantwortungsbewusst (D), • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K), 	<p><i>Beispiel:</i></p> <p>Sortieren mit Waage</p> <p>Die Schülerinnen und Schüler bekommen die Aufgabe, kleine, optisch identische Kunststoffbehälter aufsteigend nach ihrem Gewicht zu sortieren. Dazu steht ihnen eine Balkenwaage zur Verfügung, mit deren Hilfe sie das Gewicht zweier Behälter vergleichen können.</p> <p><i>Materialien:</i> Computer science unplugged – Sorting Algorithms, URL: www.csunplugged.org/sorting-algorithms (abgerufen: 30. 03. 2014)</p>
<p>2. Systematisierung von Algorithmen und Effizienzbetrachtungen</p> <p>(a) Formulierung (falls selbst gefunden) oder Erläuterung von Algorithmen in grafischer Darstellungsform (NSD oder Ablaufplan) oder als Pseudocode</p> <p>(b) Anwendung von Sortieralgorithmen auf verschiedene Beispiele</p> <p>(c) Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche</p> <p>(d) Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des</p>	<ul style="list-style-type: none"> • implementieren Algorithmen in einer Programmiersprache (I), • testen Programme schrittweise (I), • dokumentieren und strukturieren Programme (D) 	<p><i>Beispiele:</i></p> <p>Sortieren durch Auswählen, Sortieren durch Vertauschen, Quicksort</p> <p>Quicksort ist als Beispiel für einen Algorithmus nach dem Prinzip <i>Teile und Herrsche</i> gut zu behandeln. Kenntnisse in rekursiver Programmierung sind nicht erforderlich, da eine Implementierung nicht angestrebt wird.</p> <p><i>Materialien:</i></p> <p>Computer science unplugged – Sorting Algorithms, URL:</p>

<p>Speicherplatzbedarfs</p> <p>(e) Analyse eines weiteren Sortieralgorithmus</p>		<p>www.csunplugged.org/sorting-algorithms abgerufen: 30. 03. 2014</p>
<p>3. Binäre Suche auf sortierten Daten</p> <p>(a) Suchaufgaben im Alltag und im Kontext informatischer Systeme</p> <p>(b) Evtl. Simulationsspiel zum effizienten Suchen mit binärer Suche</p> <p>(c) Effizienzbetrachtungen zur binären Suche</p>		<p><i>Beispiel:</i> Simulationsspiel zur binären Suche nach Tischtennisbällen</p> <p>Mehrere Tischtennisbälle sind nummeriert, sortiert und unter Bechern verdeckt. Mit Hilfe der binären Suche kann sehr schnell ein bestimmter Tischtennisball gefunden werden.</p> <p><i>Materialien:</i></p> <p>Computer science unplugged – Searching Algorithms, URL: www.csunplugged.org/searching-algorithms, (abgerufen: 30. 03. 2014)</p>

Unterrichtsvorhaben EF-VI

Thema: Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes

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

Vorhabenbezogene Konkretisierung:

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

Anschließend wird verstärkt auf den Aspekt des Datenschutzes eingegangen. Dazu wird das Bundesdatenschutzgesetz in Auszügen behandelt und auf schülernahe Beispielsituationen zur Anwendung gebracht. Dabei steht keine formale juristische Bewertung der Beispielsituationen im Vordergrund, die im Rahmen eines Informatikunterrichts auch nicht geleistet werden kann, sondern vielmehr eine persönliche Einschätzung von Fällen im Geiste des Datenschutzgesetzes.

Zeitbedarf: 15 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Selbstständige 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 Morsen zum modernen Digitalcomputer“ • „Eine kleine Geschichte der Kryptographie: von Caesar zur Enigma“ • „Von Nullen, Einsen und mehr: Stellenwertsysteme und wie man mit ihnen rechnet“ • „Kodieren von Texten und Bildern: ASCII, RGB und mehr“ • „Auswirkungen der Digitalisierung: Veränderungen der Arbeitswelt und Datenschutz“ <p>(b) Vorstellung und Diskussion durch Schülerinnen und Schüler</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A), • erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A), • stellen ganze Zahlen und Zeichen in Binärcodes dar (D), • interpretieren Binärcodes als Zahlen und Zeichen (D), • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation. (K). 	<p><i>Beispiel:</i> Ausstellung zu informatischen Themen</p> <p>Die Schülerinnen und Schüler bereiten eine Ausstellung zu informatischen Themen vor. Dazu werden Stellwände und Plakate vorbereitet, die ggf. auch außerhalb des Informatikunterrichts in der Schule ausgestellt werden können.</p> <p><i>Materialien:</i></p> <p>Schülerinnen und Schüler recherchieren selbstständig im Internet, in der Schulbibliothek, in öffentlichen Bibliotheken, usw.</p>
<p>2. Vertiefung des Themas Datenschutz</p> <p>(a) Erarbeitung grundlegender Begriffe des Datenschutzes</p>		<p><i>Beispiel:</i> Fallbeispiele aus dem aktuellen Tagesgeschehen</p> <p>Die Schülerinnen und Schüler bearbeiten</p>

<p>(b) Problematisierung und Anknüpfung an die Lebenswelt der Schülerinnen und Schüler</p> <p>(c) Diskussion und Bewertung von Fallbeispielen aus dem Themenbereich „Datenschutz“</p>		<p>Fallbeispiele aus ihrer eigenen Erfahrungswelt oder der aktuellen Medienberichterstattung.</p> <p><i>Materialien:</i></p> <p>Materialblatt zum Bundesdatenschutzgesetz (Download EF-VI.1)</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: Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen

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

Vorhabenbezogene Konkretisierung:

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

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse Queue</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse Queue</p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Queue</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), analysieren und erläutern Algorithmen und Programme (A), beurteilen die syntaktische Korrektheit und Funktionalität von Programmen (A), ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), 	<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.</p> <p>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>2. Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse Stack</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse Stack</p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Stack</p>	<ul style="list-style-type: none"> ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), modifizieren Algorithmen und Programme (I), 	<p><i>Beispiel:</i> Heftstapel In einem Heftstapel soll das Heft einer Schülerin gefunden werden.</p> <p>oder</p> <p><i>Beispiel:</i> Kisten stapeln 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 List</p> <p>(a) Erarbeitung der Vorteile der Klasse List im Gegensatz zu den bereits bekannten linearen Strukturen</p> <p>(b) Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse List.</p>	<ul style="list-style-type: none"> implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), interpretieren Fehlermeldungen und 	<p><i>Beispiel:</i> Abfahrtslauf 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>

	korrigieren den Quellcode (I),	
<p>4. Vertiefung - Anwendungen von Listen, Stapeln oder Schlangen in mindestens einem weiteren Kontext</p>	<ul style="list-style-type: none"> • testen Programme systematisch anhand von Beispielen (I), • stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D). 	<p><i>Beispiel: Skispringen</i> 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: Terme in Postfix-Notation</i> Die sog. UPN (<i>Umgekehrt-Polnische-Notation</i>) bzw. <i>Postfix-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: Rangierbahnhof</i> Auf einem Güterbahnhof gibt es drei Gleise, die nur zu einer Seite offen sind. Wagons können also von einer Seite auf das Gleis fahren und nur rückwärts wieder hinausfahren. Die Wagons tragen Nummern, wobei die Nummer jedoch erst eingesehen werden kann, wenn der Wagon der vorderste an der offenen Gleisseite ist. (Zwischen den Wagons herumzuturnen, um die anderen Wagonnummern zu lesen, wäre zu gefährlich.) Zunächst stehen alle Wagons unsortiert auf einem Gleis. Ziel ist es, alle Wagons in ein anderes Gleis zu fahren, so dass dort die Nummern der Wagons vom Gleisende aus aufsteigend in richtiger Reihenfolge sind. Zusätzlich zu diesen beiden Gleisen gibt es ein Abstellgleis, das zum Rangieren benutzt werden kann.</p>

Unterrichtsvorhaben Q1-II:

Thema: Vertiefung 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 modellierten Datenstrukturen generalisieren?*

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 von früheren Modellierungen sinnvoll erscheint. Um die Aufgabe einzugrenzen, sollen dann einige Teile durch eigene Modellierungen oder Teile von anderen Java-Klassen unter Anpassung der Modellierungen ersetzt werden.

Die Schülerinnen und Schülern erläutern und modifizieren dabei ihre früher erstellten Modellierungen und erlernen Konzepte der Generalisierung im Rahmen der OOM. Klassen und ihre Beziehungen werden unter Beachtung von Sichtbarkeitsbereichen und den Zugriffsmethoden für Attribute sowie der Konstruktoren von Klassen notiert.

Zeitbedarf: 8 Stunden

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</p> <p>(c) Erweiterung der Modellierung um Set-/Get-Methoden und Konstruktoren</p> <p>(d) Generalisierung einer Modellierung durch</p> <p style="padding-left: 20px;">a. Trennung von Struktur und Inhalt</p> <p style="padding-left: 20px;">b. Nutzung von Subtyping und Typecasting</p> <p style="padding-left: 20px;">c. (evtl. auch Interfacelasse, Templateklasse, ...)</p> <p>(e) 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), • 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), 	<p><i>Beispiel: universelle lineare Datenstrukturen</i> Um die früher modellierten Stack, Queue, List in beliebigen Kontexten verwenden zu können, wird eine Schnittstelle entwickelt, mit der beliebige unbekannte Kassen verwaltet werden können.</p> <p>oder</p> <p><i>Beispiel: Tannenbaum</i> Ein Tannenbaum soll mit verschiedenen Arten von Schmuckstücken versehen werden, die durch unterschiedliche geometrische Objekte dargestellt werden. Es gibt Kugeln, Päckchen in der Form von Würfeln und Zuckerringe in Form von Toren. Ein Prototyp, der bereits mit Kugeln geschmückt werden kann, kann zur Verfügung gestellt werden. Da alle Schmuckstücke über die Funktion des Auf- und Abschmückens verfügen sollen, liegt es nahe, dass entsprechende Methoden in einer gemeinsamen Oberklasse realisiert werden.</p>

Unterrichtsvorhaben Q1-III:

Thema: Suchen und Sortieren auf linearen Datenstrukturen

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

Vorhabenbezogene Konkretisierung:

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

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

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

Zeitbedarf: 16 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Suchen von Daten in Listen und Arrays</p> <p>(a) Lineare Suche in Listen und in Arrays</p> <p>(b) Binäre Suche in Arrays als Beispiel für rekursives Problemlösen</p> <p>(c) Untersuchung der beiden Suchverfahren hinsichtlich ihrer Effizienz (Laufzeitverhalten, Speicherbedarf)</p>	<p>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> Karteiverwaltung Für ein Adressverwaltungsprogramm soll eine Methode zum Suchen einer Adresse geschrieben werden.</p> <p>oder</p> <p><i>Beispiel:</i> Bundesjugendspiele 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 drei Disziplinen „Lauf“, „Sprung“ und „Wurf“ beschränken. Im Wettkampfbüro wird das Ergebnis erstellt. Das Programm soll dafür zunächst den Besten einer Disziplin herausuchen können und später das gesamte Ergebnis nach gewissen Kriterien sortieren können.</p>
<p>2. Sortieren in Listen und Arrays - Entwicklung und Implementierung von iterativen und rekursiven Sortierverfahren</p> <p>(a) Entwicklung und Implementierung eines einfachen Sortierverfahrens für eine Liste</p> <p>(b) Implementierung eines einfachen Sortierverfahrens für ein Feld</p> <p>(c) Entwicklung eines rekursiven Sortierverfahren für ein Feld (z.B. Sortieren durch Mischen)</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), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen (I), 	<p><i>Beispiel:</i> Karteiverwaltung (s.o.)</p> <p>oder</p> <p><i>Beispiel:</i> Bundesjugendspiele (s.o.)</p> <p><i>Materialien:</i> (s.o.)</p>
	<ul style="list-style-type: none"> • stellen iterative und rekursive Algorithmen 	

<p>3. Untersuchung der Effizienz mind. der elementaren Sortierverfahren auf linearen Listen</p> <p>(a) Grafische Veranschaulichung der Sortierverfahren</p> <p>(b) Untersuchung der Anzahl der Vergleichsoperationen und des Speicherbedarf bei beiden Sortierverfahren</p> <p>(c) Beurteilung der Effizienz der beiden Sortierverfahren</p>	<p>umgangssprachlich und grafisch dar (D).</p>	<p><i>Beispiel:</i> Karteiverwaltung (s.o.)</p> <p>oder <i>Beispiel:</i> Bundesjugendspiele (s.o.)</p> <p><i>Materialien:</i> (s.o.)</p>
---	--	--

Unterrichtsvorhaben Q1-IV:

Thema: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

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

Vorhabenbezogene Konkretisierung:

Ausgehend von einer zu erstellenden oder einer vorhandenen Datenbank entwickeln Schülerinnen und Schüler für sie relevante Fragestellungen, die mit dem Datenbestand beantwortet werden sollen. Zur Beantwortung dieser Fragestellungen wird je nach gewähltem Weg entweder direkt die gegebene Datenbank von den Schülerinnen und Schülern analysiert und alle notwendigen Grundbegriffe für Datenbanksysteme sowie die erforderlichen SQL-Abfragen werden erarbeitet. Dabei wird zudem das Modell des Entity-Relationship-Diagramms behandelt, um die Entitäten inklusive ihrer Attribute und Relationen in dem vorgegebenen Datenbankschema darzustellen.

Im Kontext der Erstellung einer eigenen Datenbank ermitteln die Schülerinnen und Schüler zunächst in den Anwendungssituation die 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.

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

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>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 Datenbank • Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, Datenbankschema <p>(b) SQL-Abfragen</p> <ul style="list-style-type: none"> • Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL (SELECT (DISTINCT) ...FROM, WHERE, AND, OR, NOT) auf einer Tabelle • Analyse und Erarbeitung von SQL-Abfragen auf einer und mehrerer Tabelle zur Beantwortung der Fragestellungen (JOIN, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, <>, >, <, >=, <=, LIKE, BETWEEN, IN, IS NULL) <p>(c) Vertiefung an einem weiteren Datenbankbeispiel</p>	<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), • 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), 	<p><i>Beispiel:</i> VideoCenter 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. Unter http://dokumentation.videocenter.schule.de/old/video/index.html (abgerufen: 30. 03. 2014) 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 Unter www.brd.nrw.de/lerntreffs/informatik/structure/material/sek2/datenbank/en.php (abgerufen: 30. 03. 2014) wird eine Datenbank zur Verfügung gestellt, die Daten einer Schulbuch-Ausleihe 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"> • verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I), • ermitteln Ergebnisse von Datenbankabfragen über 	<p><i>Beispiel:</i> Fahrradverleih Der Fahrradverleih <i>BTR (BikesToRent)</i> verleiht unterschiedliche Typen von Fahrrädern diverser</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 Datenbankschematas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln <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>mehrere verknüpfte Tabellen (D),</p> <ul style="list-style-type: none"> • 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>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: Reederei</i> 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><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.</p> <p>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 (abgerufen: 30.03. 2014) findet man ein freies Online-Buchungssystem inklusive Demo, an Hand derer man erläutern kann, worum es in dem Projekt geht.</p>
---	---	--

Beispiel: Schulverwaltung

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.

Unterrichtsvorhaben Q1-V:

Thema: Sicherheit und Datenschutz in Netzstrukturen

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

Vorhabenbezogene Konkretisierung:

In diesem Unterrichtsvorhaben werden Zugriffe zwischen Computern, Topologien von Netzwerken, eine Client-Server-Struktur, das TCP/IP-Schichtenmodell sowie Sicherheitsaspekte beim Zugriff auf Datenbanken und verschiedene symmetrische und asymmetrische kryptografische Verfahren analysiert und erläutert. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht runden das Unterrichtsvorhaben ab.

Zeitbedarf: 10 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Daten in Netzwerken und Sicherheitsaspekte in Netzen sowie beim Zugriff auf Datenbanken</p> <p>(a) Beschreibung von Zugriffen in einem Rechnernetz anhand eines Anwendungskontextes</p> <p>(b) Beschreibung von Peer-to-Peer Netzwerken und Client-Server-Strukturen</p> <p>(c) Netztopologien und Grundlagen von Client-Server-Strukturen unter Nutzung von Ports zur Unterscheidung von Serverdiensten</p> <p>(d) Beispielhafte Client-Server-Kommunikationen unter Beachtung eines Schichtenmodells</p> <p>(e) Vertraulichkeit, Integrität, Authentizität in Netzwerken sowie symmetrische und asymmetrische kryptografische Verfahren als Methoden Daten im Netz verschlüsselt zu übertragen</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • beschreiben und erläutern Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A), • analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A), • untersuchen und bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes und die Sicherheit von Informatiksystemen sowie die Einhaltung der Datenschutzbestimmungen und des Urheberrechts (A), • untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A), • nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D). 	<p><i>Materialien:</i> Einsatz des Simulationsprogramms „Filius“ und/oder „netemul“ sowie weiterer Software zur Visualisierung des Aufbaus und der Funktionsweise von Rechner- netzen.</p> <p>Ausgehend von einem PC soll dieser in immer größere Netze integriert werden. Dabei werden die Schichten des TCP/IP-Modells oder des OSI-ISO-Modells sowie die Topologien und das Routing betrachtet.</p>
<p>2. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht</p>		<p><i>Materialien:</i></p>

Unterrichtsvorhaben Q2-I:

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:

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

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

Unter anderem sollen die verschiedenen Baumtraversierungen (Pre-, Post- und Inorder) implementiert werden. Unterschiede bezüglich der Möglichkeit, den Baum anhand der Ausgabe der Bauminhalte via Pre-, In- oder Postorder-Traversierung zu rekonstruieren, werden dabei ebenfalls angesprochen, indem die fehlende Umkehrbarkeit der Zuordnung Binärbaum \square 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: 24 Stunden

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 nicht-linearer) 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 	<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. (Dies gilt für alle Teilbäume.)</p> <p>oder</p>

	<p>Möglichkeiten der Polymorphie (M),</p> <ul style="list-style-type: none"> • 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), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren 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><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><i>oder</i></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. 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><i>Beispiel:</i> Informatikerbaum als binärer Baum</p> <p>In einem <i>binären Baum</i> werden die Namen und die Geburtsdaten von Informatikern</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>lexikographisch geordnet abgespeichert. 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. (Dies 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 <code>BinarySearchTree</code></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramm, grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften</p> <p>(c) Erarbeitung der Klasse <code>BinarySearchTree</code> und</p>		<p><i>Beispiel:</i> Informatikerbaum als <i>Suchbaum</i></p> <p>In einem binären <i>Suchbaum</i> werden die Namen und die Geburtsdaten von Informatikern lexikographisch geordnet abgespeichert. 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. (Dies gilt für alle Teilbäume.)</p> <p>Folgende Funktionalitäten werden</p>

<p>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>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</p> <p><i>Beispiel:</i> Buchindex</p> <p>Es soll eine Anwendung entwickelt werden, die anhand von Stichworten und zugehörigen Seitenzahlen ein Stichwortregister erstellt.</p> <p><i>oder</i></p> <p>Entscheidungsbäume, ...</p>

Unterrichtsvorhaben Q2-II:

Thema: Endliche Automaten und formale Sprachen

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

Vorhabenbezogene Konkretisierung:

Anhand kontextbezogener Beispiele werden Grammatiken regulärer Sprachen entwickelt, untersucht und modifiziert.

Anhand kontextbezogener Beispiele werden endliche Automaten 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.

Für Grammatiken und Automaten werden verschiedene Darstellungsformen ineinander überführt. An einem Beispiel wird ein nichtdeterministischer Akzeptor eingeführt als Alternative gegenüber einem entsprechenden deterministischen Akzeptor.

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

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p>1. Untersuchung und Entwicklung von Grammatiken regulärer Sprachen</p> <p>(a) Erarbeitung der formalen Darstellung regulärer Grammatiken</p> <p>(b) Untersuchung, Darstellung, Modifikation und Entwicklung von Grammatiken</p> <p>(c) Entwicklung von Ableitungsbäumen zur Überprüfung von Wörtern</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 Sprachen (A), zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A), 	<p><i>Beispiele:</i></p> <p>verschiedene Grammatiken für die Schafssprache bzw. ähnlicher Kunstsprachen</p> <p>oder</p> <p>reguläre Grammatik für Wörter mit ungerader Parität, Grammatik für Wörter, die bestimmte Zahlen repräsentieren, Satzgliederungsgrammatik</p>
<p>2. Endliche Automaten</p> <p>(a) Automaten im Kontext einer Anwendung unter Beachtung der gewünschten Sichtweise (Akzeptor, realer Automat, Zustandsorientierte Modellierung, Visualisierung von Grammatiken, ...)</p> <p>(b) Formalisierung zur mathematischen Beschreibung eines endlichen Automaten</p> <p>(c) Untersuchung, Darstellung und Entwicklung endlicher Automaten</p>	<ul style="list-style-type: none"> ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A), entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M), entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M), entwickeln zur akzeptierten Sprache eines Automaten die zugehörige Grammatik (M), entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M), 	<p><i>Beispiele:</i></p> <p>Bei der Entwicklung der Sprache der durch drei teilbaren Zahlen anhand einer mathematischen Analyse bietet sich eine grafische Veranschaulichung als „Automat“ an, so dass ein Übergang zu Automaten als Visualisierung von Grammatiken möglich ist.</p> <p>oder</p> <p>Roboter, Zustandsänderung eines Objekts „Auto“, Akzeptor für bestimmte Zahlen, Akzeptor für Teilwörter in längeren Zeichenketten, Akzeptor für Terme</p>
<p>3. Grenzen endlicher Automaten</p> <p>(a) Umwandlungsschema von reg. Grammatiken und</p>	<ul style="list-style-type: none"> modifizieren Grammatiken regulärer Sprachen (M), entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M), 	<p><i>Beispiele:</i></p> <p>Klammerausdrücke, $a^n b^n$ im Vergleich zu $(ab)^n$</p>

<p>DEA</p> <p>(b) Problem des Zählens bei DEA</p>	<ul style="list-style-type: none">• stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D),• ermitteln die Sprache, die ein endlicher Automat akzeptiert (D).• beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D).	
---	--	--

Unterrichtsvorhaben Q2-III:

Thema: Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

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

Vorhabenbezogene Konkretisierung:

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

Zeitbedarf: 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p>1. Von-Neumann-Architektur und die Ausführung maschinennaher Programme</p> <p>a) prinzipieller Aufbau einer von Neumann-Architektur mit CPU, Rechenwerk, Steuerwerk, Register und Hauptspeicher</p> <p>b) einige maschinennahe Befehlen und ihre Repräsentation in einem Binär-Code, der in einem Register gespeichert werden kann</p> <p>c) Analyse und Erläuterung der Funktionsweise eines einfachen maschinennahen Programms</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • erläutern die Ausführung eines einfachen maschinennahen Programms sowie die Datenspeicherung auf einer „Von-Neumann-Architektur“ (A), • untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A). 	<p><i>Beispiel:</i></p> <p><i>Rollenspiel zum VNR</i></p> <p><i>oder</i></p> <p>Addition von 4 zu einer eingegeben Zahl mit einem Rechnermodell</p>
<p>2. Grenzen der Automatisierbarkeit</p> <p>Beurteilung des Einsatzes von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen</p>		<p><i>Beispiel:</i> Halteproblem</p>

Unterrichtsvorhaben Q2-IV:

Thema: Wiederholung im Kontext eines Projektes

Leitfragen: *Welche Aspekte der bisherigen Unterrichtsinhalte stehen in direktem Zusammenhang mit dem gewählten Projekt unter Beachtung der aktuellen Zentralabiturvorgaben? Welche Erweiterungen der bisherigen Konzeptionen sind unter Beachtung der aktuellen Zentralabiturvorgaben bedeutend?*

Vorhabenbezogene Konkretisierung:

Anhand eines Wiederholungsblockes / Projektes wiederholen die Schülerinnen und Schüler die bisherigen Konzepte, implementieren Aufgaben in Java und erweitern ihre Kenntnisse um spezifische Aspekte der aktuellen Zentralabiturvorgaben.

Zeitbedarf: 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p>Wiederholungsblock / Projekte</p>	<p>(siehe oben bei den beschriebenen Kompetenzen bezüglich des im Projekt gewählten Unterrichtsvorhabens)</p>	<p>Ansatz: Wiederholungsblock als Weiterführung der OOM im Kontext der abstrakten Datentypen</p> <p>Kernspekt: Umsetzungsebene der abstrakten Datentypen auf der Basis der aktuellen Spezifikationen des Zentralabiturs</p> <p>Erweiterungen:</p> <ol style="list-style-type: none"> 1) ADT Stack, Queue und List <ul style="list-style-type: none"> - Idee des Konstruktor für die geforderte Methode "create()" - Idee der Set-/Get-Methoden für die Trennung von Schreib- und Lesezugriff - Trennung von Struktur und Inhalt für die universelle Nutzung einer Datenstruktur <ul style="list-style-type: none"> => Masterobjekt "object" als universeller Pointerdatentyp => "Subtyping, Typecasting, Interfaceklassen, Templateklassen - verschiedene Umsetzungsvarianten für die Methode „insert()“ zeigen die „Freiheiten“ des Implementierers 2) ADT Baum: Die Trennung von Struktur und Inhalt liefert ein Problem bei der Nutzung der Methode "insert()" innerhalb der Strukturebene, da die Methode Informationen der Anwendungsebene benötigt. <ul style="list-style-type: none"> - Ansatz der "abstrakten" Klassen - Erweiterung der Klassenbeziehungen um die Vererbung inklusive Polymorphie 3) ADT Graph: Die Verwaltung von 0..n Knoten durch den Graphen bedingt die Problematik der Erstellung von Klassendiagrammen mit Kardinalitäten größer als 1 <ul style="list-style-type: none"> - 1. Aspekt: Ist die Verwaltungsstruktur der Assoziationen wichtig für die aktuell zu erstellende Software? <ul style="list-style-type: none"> a) soll die Struktur der Speicherung betont werden, wird im KD ein Attribut verwendet b) soll die parallele Zugriffsmöglichkeit betont werden, wird im KD eine Assoziation der Kardinalität 0..n verwendet - 2. Aspekt: Wie entlastet man die Erstellung eines Klassendiagramms um die frühzeitige Festlegung der Speicherstruktur? <ul style="list-style-type: none"> Aufsplittung der UML-Klassendiagramme in die Entwurfs- und Implementationsdiagramme

2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

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

Überfachliche Grundsätze:

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

Fachliche Grundsätze:

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

2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

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

2.3.1 Beurteilungsbereich Klausuren

Verbindliche Absprachen:

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

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

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

Kriterien

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

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

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

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

2.3.2 Beurteilungsbereich Sonstige Mitarbeit

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

Verbindliche Absprachen der Fachkonferenz

- Alle Schülerinnen und Schüler führen in der Einführungsphase in Kleingruppen ein Kurzprojekt durch und fertigen dazu eine Arbeitsmappe mit Arbeitstagebuch an. Dies wird in die Note für die Sonstige Mitarbeit einbezogen. (siehe Unterrichtsvorhaben EII-EIII, speziell „Anfertigung einer Projektmappe“)
- In der Qualifikationsphase erstellen, dokumentieren und präsentieren die Schülerinnen und Schüler in Kleingruppen anwendungsbezogene Softwareprodukte. Diese werden in die Note für die Sonstige Mitarbeit einbezogen.

Leistungsaspekte

Mündliche Leistungen

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

Praktische Leistungen am Computer

- Implementierung, Test und Anwendung von Informatiksystemen

Sonstige schriftliche Leistungen

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

Kriterien

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

Die Bewertungskriterien stützen sich auf

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

Besonderes Augenmerk ist dabei auf

- die sachliche Richtigkeit,

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

Bei Gruppenarbeiten ist auch auf

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

Wert zu legen.

Bei Projektarbeit ist darüber hinaus auf

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

Wert zu legen.

Kategorien und Dokumentation

Basis für die fortlaufende Dokumentation der Kompetenzentwicklungen der Schülerinnen und Schüler ist ein vierstufiges System (zB -, -+, +-, +), durch das die individuellen Kompetenzausprägungen der Schülerinnen und Schüler pro Stunde für zwei ausgewählte Kategorien verdeutlicht werden. Zusätzliche Kommentierungen im Rahmen der Stundendokumentation können besondere Aspekte der Beiträge der Schülerinnen und Schüler zur Unterstützung der pädagogischen Notenfindung ergänzen. Ebenso spielen Entwicklung, Konstanz und Tendenzen eine wesentliche Rolle bei der Notenfindung, die keiner arithmetischen Berechnung entspricht.

In der folgenden Auflistung werden die Kategorien durch ihre Kernaspekte und evtl. Teilaspekte verdeutlicht und die Kompetenzausprägungen der Stufen (+) und (-) verdeutlicht. Für alle Kategorien ergeben sich die mittleren Ausprägungen (+- und -+) durch entsprechende Abstufungen.

1. Kategorie: Kooperation

Kernaspekte: Umgang mit Mitschülern und persönliches Engagement

Für ein (+) muss eine Schülerin / ein Schüler

ein hohes persönliches Engagement zeigen und die Gruppenphase durch einen konstruktiven kommunikativen Austausch unter Einbringung der Mitschülerinnen und Mitschüler gemäß deren Stärken verbessern.

Bei einem (-) fühlt sich eine Schülerin / ein Schüler

nicht für den Gruppenerfolg verantwortlich und arbeitet gegen einen konstruktiven Gruppenprozess

2. Kategorie: Unterrichtsbeiträge

Kernaspekte: Qualität, Quantität und "passive Beteiligung" (der stille, aufmerksame, mit lernende Schüler)

konkretisierende Teilaspekte:

- ständige (aktive) Teilnahme unter Nutzung der Fachsprache
- Problemerkennung mit selbständiger Lösung
- zielführende, anspruchsvolle, stichhaltige Beiträge, die den Unterricht voranbringen
- differenzierte, vorausdenkende sowie verknüpfende Beiträge zur Vernetzung der Fachinhalte

Ein (+) kennzeichnet eine Schülerin / einen Schüler,

die/der kontinuierlich aufmerksam dem Unterricht folgt und die/der an den Gelenkstellen des Unterrichtes durch ihre/seine Beiträge den Unterricht weiterbringt.

Ein (-) kennzeichnet eine Schülerin / einen Schüler,

die/der nur sporadisch dem Unterrichtsgeschehen ihre/seine Aufmerksamkeit schenkt und die/der zu keiner Weiterführung des Unterrichtes beiträgt oder diese behindert.

3. Kategorie: Selbständigkeit

(umfasst Phasen der Einzelarbeit und der arbeitsteiligen Gruppenarbeit)

Kernaspekte: Durchhaltevermögen und zielstrebiges Vorgehen

konkretisierende Teilaspekte:

- Ausdauer
- Herausforderung annehmend
- zielgerichtetes Vorgehen
- strategisches Vorgehen

Eine mit (+) gekennzeichnete Schülerin / ein mit (+) gekennzeichneter Schüler

verschafft sich einen Überblick über den Leistungsauftrag und plant sein Vorgehen unter dem Gesichtspunkt der Optimierung von Zeitmanagement, Arbeitsaufteilung und Hilfesuchung. Sie / er schreckt vor Herausforderungen nicht zurück und begegnet Widerstand mit Ehrgeiz und Tatkraft gewissenhaft.

Eine mit (-) gekennzeichnete Schülerin / ein mit (-) gekennzeichneter Schüler

zeigt ein deutliches Vermeidungsverhalten und ist in weiten Teilen des Unterrichtsgeschehens desorientiert.

4. Kategorie: Produktbewertung und Präsentation

(umfasst Aspekte der Hausaufgabenstellung und -kontrolle, der Arbeit mit Portfolios, der Projektergebnisse und der schriftlichen Lernerfolgskontrollen)

Kernaspekte: Qualität des Produktes und der Präsentation

konkretisierende Teilaspekte:

- Verständlichkeit
- Übersichtlichkeit

- Funktionstüchtigkeit
- Differenziertheit
- Quantität bezüglich Umfang und Differenziertheit

Eine mit (+) gekennzeichnete Schülerin / ein mit (+) gekennzeichneter Schüler liefert ein Produkt gemäß der vorherigen Spezifikation ab und stellt dieses situationsgemäß vor.

Bei einer / einem mit (-) gekennzeichneten Schülerin / Schüler ist das Produkt nicht den Spezifikationen entsprechend und wird nicht der unterrichtlichen Vorgaben gemäß vorgestellt.

Grundsätze der Leistungsrückmeldung und Beratung

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

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

Die Leistungsrückmeldung kann

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

erfolgen.

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

3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen

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

3.1 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, werden mögliche Zusammenarbeiten mit anderen Fächern in gemeinsamen Fachkonferenz diskutiert und festgelegt.

3.2 Vorbereitung auf die Erstellung der Facharbeit

An geeigneten Stellen werden im Unterricht fachbezogenen Hinweise zur Erstellung von Facharbeiten gegeben. Das betrifft u.a. Themenvorschläge, Hinweise zu den Anforderungen und zur Bewertung. Dabei obliegt die Wahl zugelassener Themenbereiche für die Facharbeit der betroffenen Kollegin / dem betroffenen Kollegen und wird nicht auf den Bereich der Unterrichtsthemen festgelegt.

3.3 Exkursionen und MINT-Tage

Exkursionen können abhängig von dem jeweiligen Unterrichtsgang zu verschiedenen Themenkomplexen stattfinden und werden im Unterricht vor- und nachbereitet. Die Wahl der Exkursionsorte sowie die Anzahl und Durchführung wird durch die Lehrkraft festgelegt. Die MINT-Tage finden zweimal pro Schuljahr zu zentral festgelegten Terminen statt und befassen sich mit allgemeinen Themengebieten aus dem MINT-Bereich und erweitern die unterrichtlichen Themengebiete bzw. greifen diese auf. Vor- und Nachbereitung findet auch hier im Rahmen des Unterrichtes statt.

4 Qualitätssicherung und Evaluation

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

Das schulinterne Curriculum (siehe 2.1) ist zunächst bis 2017 für den ersten Durchgang durch die gymnasiale Oberstufe nach Erlass des Kernlehrplanes verbindlich. Erstmals nach Ende der Einführungsphase im Sommer 2015, werden in einer Sitzung der Fachkonferenz Erfahrungen ausgetauscht und ggf. Änderungen für den nächsten Durchgang der Einführungsphase beschlossen, um erkannten ungünstigen Entscheidungen schnellstmöglich entgegenwirken zu können.

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