



Ville-Gymnasium der Stadt Erfstadt

Schwalbenstr. 1 · 50374 Erfstadt · Tel.: (0 22 35) 92 22 53 · Fax: (0 22 35) 92 22 55
E-Mail: Sekretariat@Ville-Gymnasium.de · Internet: www.Ville-Gymnasium.de

Schulinterner Lehrplan zum Kernlehrplan für die gymnasiale Oberstufe im Fach

Informatik

(Stand: 10.09.2015)

Inhaltsverzeichnis

1 Die Fachgruppe Informatik des Vile-Gymnasiums.....	3
2 Entscheidungen zum Unterricht.....	4
2.1 Unterrichtsvorhaben.....	4
2.1.1 Übersichtsraster Unterrichtsvorhaben.....	5
2.1.2 Konkretisierte Unterrichtsvorhaben.....	10
2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit.....	35
2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung.....	36
3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen.....	37
4 Qualitätssicherung und Evaluation.....	38

1 Die Fachgruppe Informatik des Vile-Gymnasiums

Das Fach Informatik wird am Vile-Gymnasium in der Sekundarstufe I als zweistündiger Wahlpflichtkurs in den Jahrgangsstufen 8 und 9 angeboten. Weitere Informationen hierzu sind dem schulinternen Curriculum für das Fach Informatik in der Sekundarstufe I – abrufbar unter <http://ville-gymnasium.de/unterricht/faecher> – zu entnehmen.

In der Sekundarstufe II wird das Fach Informatik als Grundkurs von der Einführungsphase bis zum Abitur angeboten, wobei die Möglichkeit besteht, Informatik als drittes oder viertes Abiturfach zu wählen. Durch die Belegung des Faches Informatik kann durch zusätzliche Wahl einer Naturwissenschaft (Biologie, Chemie oder Physik) die Pflichtbindung für einen naturwissenschaftlichen Schwerpunkt abgedeckt werden.

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

Der Unterricht der Sekundarstufe II wird unter Benutzung der Programmiersprache Java durchgeführt. Zusätzlich kommt in der Einführungsphase eine didaktische Lernumgebung zum Einsatz.

Der Grundkurs Informatik bietet für Schülerinnen und Schüler mit mathematischem und technischem Interesse die Möglichkeit, andersartige Denkstrukturen und Lösungsansätze kennen zu lernen, die über die Sichtweise des reinen Anwenders von Computertechnologie hinausgehen. In Zukunft werden solche Hintergrundkenntnisse in Entscheidungsprozessen eine immer wichtigere Rolle spielen, sowohl in der Öffentlichkeit als auch in Unternehmen.

Das Schulfach Informatik blickt hinter die Kulissen einer immer weiter voranschreitenden Technisierung und verdeutlicht Inhalte, Arbeitsweisen und Methoden, die zum tieferen Verständnis der Zusammenhänge erforderlich sind. Das Fach Informatik bemüht sich dabei, Kontexte anderer Fächer aufzugreifen und die informations- und kommunikationstechnische Seite näher zu beleuchten.

Durch den gesamten Kurs zieht sich ebenfalls die Frage, welche Auswirkungen Entwicklungen im Bereich der Informatik auf die Gesellschaft haben und wie umgekehrt die Gesellschaft versucht Einfluss auf die Informatik zu nehmen.

Zurzeit besteht die Fachschaft Informatik des Vile-Gymnasiums aus drei Lehrkräften, von denen zwei das Fach auch in der Sekundarstufe II unterrichten. Am Vile-Gymnasium stehen fünf Computerräume mit 14 bis 18 Computerarbeitsplätzen zur Verfügung, wobei sowohl Windows- als auch Mac-OS-X-Betriebssysteme zum Einsatz kommen. Alle Rechner sind jeweils in ein Rechnernetzwerk eingebunden. Die Sicherung der eigenen Daten erfolgt in Eigenverantwortung der Schülerinnen und Schüler jeweils zusätzlich auf USB-Sticks, welche die Schülerinnen und Schüler ständig verfügbar halten.

Informationen zu den allgemeinen schulischen Rahmenbedingungen finden sich auf der Homepage des Vile-Gymnasiums (<http://ville-gymnasium.de>). Insbesondere wird auf den Beitrag unter <http://ville-gymnasium.de/wir/unsere-schule> verwiesen.

2 Entscheidungen zum Unterricht

2.1 Unterrichtsvorhaben

Die Fachkonferenz Informatik hat mit diesem schulinternen Lehrplan sieben Unterrichtsvorhaben zur Durchführung in der Einführungsphase und sieben Unterrichtsvorhaben zur Durchführung in der Qualifikationsphase festgelegt. Die Unterrichtsvorhaben der Qualifikationsphase beziehen sich dabei auf den Grundkurs. Zurzeit wird am Ville-Gymnasium kein Leistungskurs angeboten.

Diese verbindlich durchzuführenden Unterrichtsvorhaben sind in Kapitel 2.1.1 aufgelistet. Der dort angegebene Zeitbedarf dient als grober Orientierungsrahmen, der nach Bedarf über- oder unterschritten werden darf. Um Freiraum für Vertiefungen, zusätzliche Unterrichtsvorhaben (z.B. besondere Interessen der Schülerinnen und Schüler, aktuelle Themen, Abiturtraining, ...) oder Besonderheiten (z.B. Berufsorientierungspraktikum, Studienfahrt, Projekttag, ...) zu erhalten, beträgt der für alle Unterrichtsvorhaben zusammen eingeplante Zeitbedarf nur ca. 75% der Bruttounterrichtszeit.

In Kapitel 2.1.1 sind zum einen verbindlich für die einzelnen Unterrichtsvorhaben diejenigen Kompetenzbereiche, Inhaltsfelder und inhaltlichen Schwerpunkte des Kernlehrplans gekennzeichnet, die im Rahmen der Durchführung des Unterrichtsvorhabens relevant zu berücksichtigen sind. Es steht der jeweiligen Fachlehrkraft frei, innerhalb eines Unterrichtsvorhabens – z.B. bei Vertiefung desselben – auch zusätzliche, nicht gekennzeichnete Kompetenzbereiche, Inhaltsfelder oder inhaltliche Schwerpunkte zu berücksichtigen.

Die in Kapitel 2.1.1 angegebene Nummerierung bringt die Unterrichtsvorhaben in eine sinnvolle, aber nicht verbindliche Reihenfolge.

In Kapitel 2.1.2 ist zu jedem Unterrichtsvorhaben eine Konkretisierung zur möglichen Umsetzung im Unterricht vorgeschlagen. Die tatsächliche Umsetzung im Unterricht obliegt jedoch der pädagogischen Freiheit der jeweiligen Fachlehrkraft, die an die Vorschläge in Kapitel 2.1.2 nicht gebunden ist. Die Umsetzung der vorgeschlagenen Konkretisierungen stellt allerdings sicher, dass insgesamt **alle** übergeordneten und konkretisierten Kompetenzerwartungen des Kernlehrplans Berücksichtigung finden. Bei Abweichung von den Vorschlägen in Kapitel 2.1.2 bleibt natürlich die Verantwortung der Fachlehrkraft bestehen, den Kernlehrplan zu beachten.

Zum anderen sind in Kapitel 2.1.1 für die einzelnen Unterrichtsvorhaben diejenigen übergeordneten und konkretisierten Kompetenzen gekennzeichnet, die durch die in Kapitel 2.1.2 vorgeschlagene Konkretisierung im Rahmen des Unterrichtsvorhabens entwickelt werden können. Die Zuordnung der übergeordneten und konkretisierten Kompetenzen zu den Unterrichtsvorhaben ist nicht verbindlich. Sicherzustellen ist jedoch, dass jede der genannten Kompetenzen im Rahmen mindestens eines – auch zusätzlichen – Unterrichtsvorhabens der jeweiligen Phase (Einführungs- bzw. Qualifikationsphase) entwickelt wird.

Mit dem „Entwickeln einer Kompetenz“ ist hierbei gemeint, für die Schülerinnen und Schüler eine Lerngelegenheit zu schaffen, die ihnen – in der Regel auf der Basis bereits zuvor entwickelter anderer Kompetenzen – bei entsprechender aktiver Mitarbeit ermöglicht, die jeweilige Kompetenzerwartung erfüllen zu können.

2.1.1 Übersichtsraster Unterrichtsvorhaben

Einführungsphase	
<p><u>Unterrichtsvorhaben EF-I</u></p> <p>Thema: <i>Was macht Informatik? - Einführung in die Inhaltsfelder der Informatik</i></p> <p>Kompetenzbereiche:</p> <ul style="list-style-type: none">- Kommunizieren und Kooperieren- Darstellen und Interpretieren- Argumentieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none">- Informatiksysteme- Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none">- Einsatz, Nutzung und Aufbau von Informatiksystemen- Wirkung der Automatisierung <p>Zeitbedarf: 6 Stunden</p>	<p><u>Unterrichtsvorhaben EF-II</u></p> <p>Thema: <i>Grundlagen der objektorientierten Analyse, Modellierung und Implementierung</i></p> <p>Kompetenzbereiche:</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- Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none">- Objekte und Klassen- Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: 9 Stunden</p>
<p><u>Unterrichtsvorhaben EF-III</u></p> <p>Thema: <i>Algorithmische Grundstrukturen in Java</i></p> <p>Kompetenzbereiche:</p> <ul style="list-style-type: none">- Argumentieren- Modellieren- Implementieren- 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- Syntax und Semantik einer Programmiersprache- Analyse, Entwurf und Implementierung einfacher Algorithmen <p>Zeitbedarf: 18 Stunden</p>	<p><u>Unterrichtsvorhaben EF-IV</u></p> <p>Thema: <i>Das ist die digitale Welt! – Einführung in die Grundlagen, Anwendungsgebiete und Verarbeitung binärer Codierung</i></p> <p>Kompetenzbereiche:</p> <ul style="list-style-type: none">- Kommunizieren und Kooperieren- Darstellen und Interpretieren- Argumentieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none">- Informatiksysteme- Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none">- Binäre Codierung und Verarbeitung- Besondere Eigenschaften der digitalen Speicherung und Verarbeitung von Daten <p>Zeitbedarf: 9 Stunden</p>

<p><u>Unterrichtsvorhaben EF-V</u></p> <p>Thema: <i>Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand lebensnaher Anforderungsbeispiele</i></p> <p>Kompetenzbereiche:</p> <ul style="list-style-type: none"> - Kommunizieren und Kooperieren - Darstellen und Interpretieren - Argumentieren - Modellieren - Implementieren <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 - Syntax und Semantik einer Programmiersprache - Analyse, Entwurf und Implementierung einfacher Algorithmen <p>Zeitbedarf: 18 Stunden</p>	<p><u>Unterrichtsvorhaben EF-VI</u></p> <p>Thema: <i>Such- und Sortieralgorithmen</i></p> <p>Kompetenzbereiche:</p> <ul style="list-style-type: none"> - Argumentieren <ul style="list-style-type: none"> • - Modellieren • - Darstellen und Interpretieren • - Kommunizieren und Kooperieren • <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Algorithmen - Daten und ihre Strukturierung <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • - Algorithmen zum Suchen und Sortieren <ul style="list-style-type: none"> - Analyse, Entwurf und Implementierung einfacher Algorithmen <ul style="list-style-type: none"> • - Objekte und Klassen • <p>Zeitbedarf: 12 Stunden</p>
<p><u>Unterrichtsvorhaben EF-VII</u></p> <p>Thema: <i>Leben in der digitalen Welt – Immer mehr Möglichkeiten und immer mehr Gefahren!?</i></p> <p>Kompetenzbereiche:</p> <ul style="list-style-type: none"> - Kommunizieren und Kooperieren - Darstellen und Interpretieren - Argumentieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Informatiksysteme - Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Geschichte der automatischen Datenverarbeitung - Wirkungen der Automatisierung - Dateisystem <p>Zeitbedarf: 6 Stunden</p>	<p style="text-align: center;">Summe Einführungsphase: 78 Stunden</p>

Qualifikationsphase - Q1

<p><u>Unterrichtsvorhaben Q1-I</u></p> <p>Thema: Wiederholung und Vertiefung der objektorientierten Modellierung</p> <p>Kompetenzbereiche:</p> <ul style="list-style-type: none"> - Modellieren - Darstellen und Interpretieren - Implementieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Daten und ihre Strukturierung - Algorithmen - Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Objekte und Klassen - Wirkung der Automatisierung <p>Zeitbedarf: 15 Stunden</p>	<p><u>Unterrichtsvorhaben Q1-II</u></p> <p>Thema: Organisation und Verarbeitung von Daten I – Modellierung und Implementierung von Anwendungen mit dynamischen und linearen Datenstrukturen</p> <p>Kompetenzbereiche:</p> <ul style="list-style-type: none"> - Modellieren - Implementieren - Darstellen und Interpretieren - Argumentieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Daten und ihre Strukturierung - Algorithmen <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Objekte und Klassen - Syntax und Semantik einer Programmiersprache - Analyse, Entwurf und Implementierung von Algorithmen - Algorithmen in ausgewählten informatischen Kontexten <p>Zeitbedarf: 21 Stunden</p>
<p><u>Unterrichtsvorhaben Q1-III</u></p> <p>Thema: Algorithmen zum Suchen und Sortieren auf linearen Datenstrukturen</p> <p>Kompetenzbereiche:</p> <ul style="list-style-type: none"> - Argumentieren - Darstellen und Interpretieren - Modellieren - Implementieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Algorithmen - Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Analyse, Entwurf und Implementierung von Algorithmen - Algorithmen in ausgewählten informatischen Kontexten - Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: 21 Stunden</p>	<p><u>Unterrichtsvorhaben Q1-IV</u></p> <p>Thema: Automaten und formale Sprachen</p> <p>Kompetenzbereiche:</p> <ul style="list-style-type: none"> - Argumentieren - Darstellen und Interpretieren - Modellieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Formale Sprachen und Automaten - Informatiksysteme <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Syntax und Semantik einer Programmiersprache - Endliche Automaten - Grammatiken regulärer Sprachen - Möglichkeiten und Grenzen von Automaten und formalen Sprachen <p>Zeitbedarf: 21 Stunden</p>

**Summe Qualifikationsphase 1:
78 Stunden**

Qualifikationsphase - Q2

Unterrichtsvorhaben Q2-I

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

Kompetenzbereiche:

- Argumentieren
- Darstellen und Interpretieren
- Modellieren
- Implementieren
- 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: 21 Stunden

Unterrichtsvorhaben Q2-II

Thema: Aufbau von und Kommunikation in Netzwerken

Kompetenzbereiche:

- 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: 15 Stunden

Unterrichtsvorhaben Q2-III

Thema: Nutzung und Modellierung von relationalen Datenbanken in Anwendungskontexten

Kompetenzbereiche:

- 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: 21 Stunden

**Summe Qualifikationsphase 2:
57 Stunden**

EF	Einführung in die Informatik (6 UStd.)	Grundlagen der Objektorientierung (9 UStd.)	Algorithmische Grundstrukturen in Java (18 UStd.)	Digitale Welt – Einführung in die binäre Codierung (9 UStd.)
	Klassen- und Objektbeziehungen (18 UStd.)	Such- und Sortieralgorithmen (12 UStd.)	Möglichkeiten und Gefahren der digitalen Welt (6 UStd.)	
Q1	Vertiefung der objektorientierten Modellierung (15 UStd.)	Dynamische und Lineare Datenstrukturen (Stack, Queue, List) (21 UStd.)	Algorithmen zum Suchen und Sortieren (21 UStd.)	Automaten und formale Sprachen (21 UStd.)
Q2	Nicht-lineare Datenstrukturen (Binärbäume) (21 UStd.)	Kommunikation in Netzwerken (15 UStd.)	Relationale Datenbanken (21 UStd.)	

2.1.2 Konkretisierte Unterrichtsvorhaben

Unterrichtsvorhaben EF-I

Thema: Was macht Informatik? – Einführung in die Inhaltsfelder der Informatik

Leitfragen: Was macht Informatik? Welche fundamentalen Konzepte müssen Informatikerinnen und Informatiker in ihre Arbeit einbeziehen, damit informatische Systeme effizient und zuverlässig arbeiten können? Wo lassen sich diese Konzepte (in Ansätzen) in dem schuleigenen Netzwerk- und Computersystem wiederfinden?

Vorhabenbezogene Konkretisierung:

Im ersten Unterrichtsvorhaben werden die fünf Inhaltsfelder des Faches Informatik beispielhaft an einem Informatiksystem erarbeitet. Das Unterrichtsvorhaben ist so strukturiert, dass die Schülerinnen und Schüler anhand bekannter Alltagstechnik die Grundideen fundamentaler informatischer Konzepte (Inhaltsfelder) größtenteils selbstständig erarbeiten und nachvollziehen.

Ausgehend von dem bekannten Bedienungs- und Funktionalitätswissen eines Navigationsgerätes werden die Strukturierung von Daten, das Prinzip der Algorithmik, die Eigenheit formaler Sprachen, die Kommunikationsfähigkeit von Informatiksystemen und die positiven und negativen Auswirkungen auf Mensch und Gesellschaft thematisiert. Das am Beispiel Navigationsgerät erworbene Wissen kann auf weitere den Schülerinnen und Schülern bekannte Informatiksysteme übertragen werden.

In einem letzten Schritt kann ausgehend von den Inhaltsfeldern das Schulnetzwerk in Ansätzen so analysiert werden, dass ein kompetenter Umgang mit diesem ermöglicht wird.

Zeitbedarf: 6 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Informatiksysteme und ihr genereller Aufbau (a) Daten und ihre Strukturierung (b) Algorithmen (c) Formale Sprachen und Automaten (d) Informatiksysteme (e) Informatik, Mensch und Gesellschaft	Die Schülerinnen und Schüler - bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A), - nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D).	Kapitel 1 Was macht Informatik? Als Anschauungsmaterial bieten sich Navigationsgeräte und Landkarten an.
2. Der kompetente Umgang mit dem Schulnetzwerk (a) Erstellen und Anlegen von Ordnerstrukturen (b) Sortieren von Dateien und Ordnern (c) Eingabe von Befehlen über Eingabeaufforderung/Terminal (d) Einzelrechner und Netzwerk (e) Sicherheit und Datenschutz		Kapitel 1 Was macht Informatik? Interview mit dem Netzwerkadministrator, Benutzer- und Datenschutzbestimmungen der Schule

Unterrichtsvorhaben EF-II

Thema: Grundlagen der objektorientierten Analyse, Modellierung und Implementierung

Leitfragen: *Wie lassen sich Gegenstandsbereiche informatisch modellieren und in einem Greenfoot-Szenario informatisch realisieren?*

Vorhabenbezogene Konkretisierung:

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

Dazu werden zunächst konkrete Gegenstandsbereiche aus der Lebenswelt der Schülerinnen und Schüler analysiert und im Sinne des objektorientierten Paradigmas strukturiert. Dabei werden die grundlegenden Begriffe der Objektorientierung und Modellierungswerkzeuge wie Objektdiagramme und Klassendiagramme eingeführt.

Im Anschluss wird die objektorientierte Analyse für das Greenfoot-Szenario Planetenerkundung durchgeführt. Die vom Szenario vorgegebenen Klassen werden von Schülerinnen und Schülern in Teilen analysiert und entsprechende Objekte anhand einfacher Problemstellungen erprobt. Die Lernenden implementieren und testen einfache Programme. Die Greenfoot-Umgebung ermöglicht es, Beziehungen zwischen Klassen zu einem späteren Zeitpunkt (Kapitel 5) zu thematisieren. So kann der Fokus hier auf Grundlagen wie der Unterscheidung zwischen Klasse und Objekt, Attribute, Methoden, Objektidentität und Objektzustand gelegt werden.

Da in Kapitel 2 zudem auf die Verwendung von Kontrollstrukturen verzichtet wird und der Quellcode aus einer rein linearen Sequenz besteht, ist auf diese Weise eine Fokussierung auf die Grundlagen der Objektorientierung möglich, ohne dass algorithmische Probleme ablenken. Natürlich kann die Arbeit an diesen Projekten unmittelbar zum nächsten Unterrichtsvorhaben (Kapitel 3) führen. Dort stehen Kontrollstrukturen im Mittelpunkt.

Zeitbedarf: 9 Stunden

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Identifikation von Objekten und Klassen</p> <p>(a) An einem lebensweltnahen Beispiel werden Objekte und Klassen im Sinne der objektorientierten Modellierung eingeführt.</p> <p>(b) Objekte werden durch Objektdiagramme, Klassen durch Klassendiagramme dargestellt.</p> <p>(c) Die Modellierungen werden einem konkreten Anwendungsfall entsprechend angepasst.</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> - ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften und ihre Operationen (M), - stellen den Zustand eines Objekts dar (D), - modellieren Klassen mit ihren Attributen und ihren Methoden (M), - implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), - implementieren Klassen in einer Programmiersprache, auch unter Nutzung dokumentierter Klassenbibliotheken (I). 	<p>Kapitel 2 Einführung in die Objektorientierung</p> <p>2.1 Objektorientierte Modellierung</p>
<p>2. Analyse von Objekten und Klassen im Greenfoot-Szenario</p> <p>(a) Schritte der objektorientierten Analyse, Modellierung und Implementation</p> <p>(b) Analyse und Erprobung der Objekte im Greenfoot-Szenario</p>		<p>Kapitel 2 Einführung in die Objektorientierung</p> <p>2.2 Das Greenfoot-Szenario „Planetenerkundung“</p> <p>Von der Realität zu Objekten</p> <p>Von den Objekten zu Klassen</p> <p>Klassendokumentation</p> <p>Objekte inspizieren</p> <p>Methoden aufrufen</p> <p>Objektidentität und Objektzustand</p>
<p>3. Implementierung einfacher Aktionen in Greenfoot</p> <p>(a) Quelltext einer Java-Klasse</p> <p>(b) Implementation eigener Methoden, Dokumentation mit JavaDoc</p> <p>(c) Programme übersetzen (Aufgabe des Compilers) und testen</p>		<p>Kapitel 2 Einführung in die Objektorientierung</p> <p>2.3 Programmierung in Greenfoot</p> <p>Methoden schreiben</p> <p>Programme übersetzen und testen</p>

Unterrichtsvorhaben EF-III

Thema: Algorithmische Grundstrukturen in Java

Leitfragen: *Wie lassen sich Aktionen von Objekten flexibel realisieren?*

Vorhabenbezogene Konkretisierung:

Das Ziel dieses Unterrichtsvorhabens besteht darin, das Verhalten von Objekten flexibel zu programmieren. Ein erster Schwerpunkt liegt dabei auf der Erarbeitung von Kontrollstrukturen. Die Strukturen Wiederholung und bedingte Anweisung werden an einfachen Beispielen eingeführt und anschließend anhand komplexerer Problemstellungen erprobt. Da die zu entwickelnden Algorithmen zunehmend umfangreicher werden, werden systematische Vorgehensweisen zur Entwicklung von Algorithmen thematisiert.

Ein zweiter Schwerpunkt des Unterrichtsvorhabens liegt auf dem Einsatz von Variablen. Beginnend mit lokalen Variablen, die in Methoden und Zählschleifen zum Einsatz kommen, über Variablen in Form von Parametern und Rückgabewerten von Methoden, bis hin zu Variablen, die die Attribute einer Klasse realisieren, lernen die Schülerinnen und Schüler die unterschiedlichen Einsatzmöglichkeiten des Variablenkonzepts anzuwenden.

Zeitbedarf: 18 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Algorithmen (a) Wiederholungen (While-Schleife, Zählschleife) (b) bedingte Anweisungen (c) Verknüpfung von Bedingungen durch die logischen Funktionen UND, ODER und NICHT (d) Systematisierung des Vorgehens zur Entwicklung von Algorithmen zur Lösung komplexerer Probleme	Die Schülerinnen und Schüler - analysieren und erläutern einfache Algorithmen und Programme (A), - entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M), - ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen zu (M), - modifizieren einfache Algorithmen und Programme (I), - implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I), - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), - implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), - testen Programme schrittweise anhand von Beispielen (I), - interpretieren Fehlermeldungen und korrigieren den Quellcode (I).	Kapitel 3 Algorithmen 3.1 Kontrollstrukturen 3.2 Wiederholungen 3.3 Zählschleifen 3.4 Bedingte Anweisungen 3.5 Logische Operationen 3.6 Algorithmen entwickeln
2. Variablen und Methoden (a) Implementierung eigener Methoden mit lokalen Variablen, auch zur Realisierung einer Zählschleife (b) Implementierung eigener Methoden mit Parameterübergabe und/oder Rückgabewert (c) Implementierung von Kontrollstrukturen (d) Realisierung von Attributen		Kapitel 4 Variablen und Methoden 4.1 Variablen 4.2 Methoden

Unterrichtsvorhaben EF-IV

Thema: Das ist die digitale Welt! – Einführung in die Grundlagen, Anwendungsgebiete und Verarbeitung binärer Codierung

Leitfragen: Wie werden binäre Informationen gespeichert und wie können sie davon ausgehend weiter verarbeitet werden? Wie unterscheiden sich analoge Medien und Geräte von digitalen Medien und Geräten? Wie ist der Grundaufbau einer digitalen Rechenmaschine?

Vorhabenbezogene Konkretisierung:

Das Unterrichtsvorhaben hat die binäre Speicherung und Verarbeitung sowie deren Besonderheiten zum Inhalt.

Im ersten Schritt erarbeiten die Schülerinnen und Schüler anhand ihnen bekannter technischer Gegenstände die Gemeinsamkeiten, Unterschiede und Besonderheiten der jeweiligen analogen und digitalen Version. Nach dieser ersten grundlegenden Einordnung des digitalen Prinzips wenden die Schülerinnen und Schüler das Binäre als Zahlensystem mit arithmetischen und logischen Operationen an und codieren Zeichen binär.

Zum Abschluss soll der grundlegende Aufbau eines Rechnersystems im Sinne der Von-Neumann-Architektur erarbeitet werden.

Zeitbedarf: 9 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Analoge und digitale Aufbereitung und Verarbeitung von Daten (a) Erarbeitung der Unterschiede von analog und digital (b) Zusammenfassung und Bewertung der technischen Möglichkeiten von analog und digital	Die Schülerinnen und Schüler - bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A), - stellen ganze Zahlen und Zeichen in Binärcodes dar (D), - interpretieren Binärcodes als Zahlen und Zeichen (D), - beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A), - nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K), - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I).	Die digitale Welt 001 – Von analog zu digital
2. Der Umgang mit binärer Codierung von Informationen (a) Das binäre (und hexadezimale) Zahlensystem (b) Binäre Informationsspeicherung (c) Binäre Verschlüsselung (d) Implementation eines Binärumrechners		Die digitale Welt 010 – Binäre Welt
3. Aufbau informatischer Systeme (a) Identifikation des EVA-Prinzips als grundlegende Arbeitsweise informatischer Systemen (b) Nachvollziehen der von-Neumann-Architektur als relevantes Modell der Umsetzung des EVA-Prinzips		Die digitale Welt 011 – Der Von-Neumann-Rechner

Unterrichtsvorhaben EF-V

Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand lebensnaher Anforderungsbeispiele

Leitfragen: Wie werden realistische Systeme anforderungsspezifisch reduziert, als Entwurf modelliert und implementiert? Wie kommunizieren Objekte und wie wird dieses dargestellt und realisiert?

Vorhabenbezogene Konkretisierung:

Das Unterrichtsvorhaben hat die Entwicklung von Objekt- und Klassenbeziehungen zum Schwerpunkt. Dazu werden, ausgehend von der Realität, über Objektidentifizierung und Entwurf bis hin zur Implementation kleine Softwareprodukte in Teilen oder ganzheitlich erstellt.

Zuerst identifizieren die Schülerinnen und Schüler Objekte und stellen diese dar. Aus diesen Objekten werden Klassen und ihre Beziehungen in Entwurfsdiagrammen erstellt.

Nach diesem ersten Modellierungsschritt werden über Klassendokumentationen und der Darstellung von Objektkommunikationen anhand von Sequenzdiagrammen Implementationsdiagramme entwickelt. Danach werden die Implementationsdiagramme unter Berücksichtigung der Klassendokumentationen in Javaklassen programmiert. In einem letzten Schritt wird das Konzept der Vererbung sowie seiner Vorteile erarbeitet.

Schließlich sind die Schülerinnen und Schüler in der Lage, eigene kleine Softwareprojekte zu entwickeln. Ausgehend von der Dekonstruktion und Erweiterung eines Spiels wird ein weiteres Projekt von Grund auf modelliert und implementiert. Dabei können arbeitsteilige Vorgehensweisen zum Einsatz kommen. In diesem Zusammenhang wird auch das Erstellen von graphischen Benutzeroberflächen eingeführt.

Zeitbedarf: 18 Stunden

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Umsetzung von Anforderungen in Entwurfsdiagramme</p> <p>(a) Aus Anforderungsbeschreibungen werden Objekte mit ihren Eigenschaften identifiziert.</p> <p>(b) Gleichartige Objekte werden in Klassen (Entwurf) zusammengefasst und um Datentypen und Methoden erweitert.</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> - analysieren und erläutern eine objektorientierte Modellierung (A), - stellen die Kommunikation zwischen Objekten grafisch dar (M), - ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), 	<p>Kapitel 5 Klassenentwurf</p> <p>5.1 Von der Realität zum Programm</p> <p>5.2 Objekte</p> <p>5.3 Klassen und Beziehungen entwerfen</p>
<p>2. Implementationsdiagramme als erster Schritt der Programmierung</p> <p>(a) Erweiterung des Entwurfsdiagramms um Konstruktoren und get- und set-Methoden</p> <p>(b) Festlegung von Datentypen in Java, sowie von Rückgaben und Parametern</p> <p>(c) Entwicklung von Klassendokumentationen</p> <p>(d) Erstellung von Sequenzdiagrammen als Vorbereitung für die Programmierung</p>	<ul style="list-style-type: none"> - modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), - ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M), - ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M), - modellieren Klassen unter Verwendung von Vererbung (M), - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), 	<p>Kapitel 5 Klassenentwurf</p> <p>5.4 Klassen und Beziehungen implementieren</p>
<p>3. Programmierung anhand der Dokumentation und des Implementations- und Sequenzdiagrammes</p> <p>(a) Klassen werden in Java-Quellcode umgesetzt.</p> <p>(b) Das Geheimnisprinzip wird umgesetzt.</p> <p>(c) Einzelne Klassen und das Gesamtsystem werden anhand der Anforderungen und Dokumentationen auf ihre Korrektheit überprüft.</p>	<ul style="list-style-type: none"> - testen Programme schrittweise anhand von Beispielen (I), - interpretieren Fehlermeldungen und korrigieren den Quellcode (I), - analysieren und erläutern einfache Algorithmen und Programme (A), - modifizieren einfache Algorithmen und Programme (I), - entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M), - stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), - dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D). 	<p>Kapitel 5 Klassenentwurf</p> <p>5.4 Klassen und Beziehungen implementieren</p>
<p>4. Vererbungsbeziehungen</p> <p>(a) Das Grundprinzip der Vererbung wird erarbeitet.</p> <p>(b) Vorteile der Vererbungsbeziehungen</p> <p>(c) Vererbung wird implementiert.</p>	<ul style="list-style-type: none"> - stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), - dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D). 	<p>Kapitel 5 Klassenentwurf</p> <p>5.5 Vererbung</p>
<p>5. Softwareprojekt, z.B.:</p> <p>(a) Analyse und Dekonstruktion eines Spiels (Modelle, Quelltexte)</p> <p>(b) Erweiterung des Spiels um weitere Funktionalitäten</p> <p>(c) Modellierung eines Spiels aufgrund einer Anforderungsbeschreibung, inklusive einer grafischen Benutzeroberfläche</p>		<p>Kapitel 7 Softwareprojekte</p> <p>7.1 Softwareentwicklung</p> <p>7.2 Oberflächen</p>

Unterrichtsvorhaben EF-VI

Thema: Such- und Sortieralgorithmen

Leitfragen: *Wie können Objekte bzw. Daten effizient gesucht und sortiert werden?*

Vorhabenbezogene Konkretisierung:

Dieses Unterrichtsvorhaben beschäftigt sich mit der Erarbeitung von Such- und Sortieralgorithmen. Der Schwerpunkt des Vorhabens liegt dabei auf den Algorithmen selbst und nicht auf deren Implementierung in einer Programmiersprache, auf die in diesem Vorhaben vollständig verzichtet werden kann.

Zunächst lernen die Schülerinnen und Schüler das Feld als eine erste Datensammlung kennen. Alle Algorithmen dieses Kapitels arbeiten auf einem Feld.

Die Schülerinnen und Schüler lernen zunächst Strategien des Suchens (lineare Suche, binäre Suche, Hashing) und dann des Sortierens (Selection Sort, Insertion Sort, Bubble Sort) kennen.

Daran anschließend lernen die Schülerinnen und Schüler zunächst Strategien des Suchens (lineare Suche, binäre Suche, Hashing) und dann des Sortierens (Selection Sort, Insertion Sort, Bubble Sort) kennen. Die Projekteinstiege dienen dazu, die jeweiligen Strategien handlungsorientiert zu erkunden und intuitive Effizienzbetrachtungen der Suchalgorithmen vorzunehmen.

Schließlich wird die Effizienz unterschiedlicher Sortierverfahren beurteilt.

Zeitbedarf: 12 Stunden

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Modellierung und Implementation von Datenansammlungen</p> <p>(a) Modellierung von Attributen als Felder</p> <p>(b) Deklaration, Instanziierung und Zugriffe auf ein Feld</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> - analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D) - entwerfen einen weiteren Algorithmus zum Sortieren (M) - beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeit und Speicherplatzbedarf (A) - ordnen Attributen lineare Datenansammlungen zu (M) 	<p>Kapitel 6 Suchen und Sortieren</p> <p>6.1 Speichern mit Struktur – Arrays</p>
<p>2. Explorative Erarbeitung von Suchverfahren</p> <p>(a) Erkundung von Strategien für das Suchen auf unsortierten Daten, auf sortierten Daten und mithilfe einer Berechnungsfunktion</p> <p>(b) Vergleich der drei Verfahren durch intuitive Effizienzbetrachtungen</p>		<p>Kapitel 6 Suchen und Sortieren</p> <p>Projekteinstieg 1: Suchen</p> <p>6.2 Suchen mit System</p> <p>Lineare Suche</p> <p>Binäre Suche</p> <p>Hashing</p>
<p>3. Systematisierung von Algorithmen und Effizienzbetrachtungen</p> <p>(a) Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im 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) Analyse eines weiteren Sortieralgorithmus (sofern nicht in (a) bereits geschehen)</p>		<p>Kapitel 6 Suchen und Sortieren</p> <p>Projekteinstieg 2: Sortieren</p> <p>6.3 Ordnung ist das halbe Leben!? – Sortieren</p> <p>Sortieren</p> <p>Selection Sort</p> <p>Insertion Sort</p> <p>Bubble Sort</p>

Unterrichtsvorhaben EF-VII

Thema: Leben in der digitalen Welt – Immer mehr Möglichkeiten und immer mehr Gefahren!?

Leitfragen: Welche Entwicklungen, Ideen und Erfindungen haben zur heutigen Informatik geführt? Welche Auswirkungen hat die Informatik für das Leben des modernen Menschen?

Vorhabenbezogene Konkretisierung:

Das Unterrichtsvorhaben stellt die verschiedenen Entwicklungsstränge der Informatik in den Fokus. Darüber hinaus wird beispielhaft analysiert und bewertet, welche Möglichkeiten und Gefahren die moderne Informationsverarbeitung mit sich bringt.

Im ersten Schritt des Unterrichtsvorhabens wird anhand von Themenkomplexen entscheidende Entwicklungen der Informatik erarbeitet. Dabei werden auch übergeordnete Tendenzen identifiziert.

Ausgehend von dieser Betrachtung kann die aktuelle Informatik hinsichtlich ihrer Leistungsfähigkeit analysiert werden. Dabei soll herausgestellt werden, welche positiven und negativen Folgen Informatiksysteme mit sich bringen können.

Zeitbedarf: 6 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Schriftzeichen, Rechenmaschine, Computer (a) Anhand von Schwerpunkten, wie z.B. Datenspeicherung, Maschinen, Vernetzung sollen wichtige Entwicklungen der Informatik vorgestellt werden. (b) Anhand der unterschiedlichen Schwerpunkte sollen universelle Tendenzen der Entwicklung der Informationsverarbeitung erarbeitet werden.	Die Schülerinnen und Schüler - bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A), - erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A), - nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K).	<i>Die digitale Welt 100 – Von der Schrift zum Smartphone</i>
2. Die Informationsverarbeitung und ihre Möglichkeiten und Gefahren (a) Ausgehend von 1. werden Tendenzen der Entwicklung der Informatik erarbeitet. (b) Informatik wird als Hilfswissenschaft klassifiziert, die weit über ihren originären Bereich hinaus Effizienz- und Leistungssteigerungen erzeugt. (c) Anhand von Fallbeispielen werden technische und organisatorische Vorteile, sowie deren datenschutzrechtlichen Nachteile betrachtet.		<i>Die digitale Welt 100 – Das Leben in der digitalen Welt</i>

Unterrichtsvorhaben Q1-I

Thema: Wiederholung und Vertiefung der objektorientierten Modellierung

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

Vorhabenbezogene Konkretisierung:

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

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

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

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

Zeitbedarf: 15 Stunden

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Wiederholung der grundlegenden Konzepte der objektorientierten Programmierung</p> <p>a) Sichtweise der objektorientierten Informatik auf die Welt</p> <p>b) OOP als informatikspezifische Modellierung der Realität</p> <p>c) Schritte der Softwareentwicklung</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> - analysieren und erläutern objektorientierte Modellierungen (A), - modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), - ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), - modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), 	<p>2.1 Die Welt ist voller Objekte ProjektEinstieg: Klassentwurf – step by step</p>
<p>2. Erweiterung der objektorientierten Programmierung</p> <p>a) Umsetzung einer Anforderung in Entwurfs- und Klassendiagramm</p> <p>b) Objektkommunikation im Sequenzdiagramm</p> <p>c) Klassendokumentation</p> <p>d) Umsetzung von Teilen der Modellierung</p>	<ul style="list-style-type: none"> - 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), - stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), - dokumentieren Klassen (D), 	<p>2.2 Gut geplant – Klassentwurf</p> <p>2.3 Vererbungshierarchien nutzen</p>
<p>3. Mensch und Technik</p> <p>a) Verantwortung von Informatikern</p> <p>b) Automatisierung des Alltags durch Informatik</p>	<ul style="list-style-type: none"> - stellen die Kommunikation zwischen Objekten grafisch dar (D), - untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A), 	<p>Die digitale Welt 001 - Mensch und Technik</p>
<p>4. Übung und Vertiefung der OOM / OOP</p>	<ul style="list-style-type: none"> - 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). 	<p>Prüfungsvorbereitung</p>

Unterrichtsvorhaben Q1-2

Thema:

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

Leitfragen:

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

Vorhabensbezogene Konkretisierung:

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

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

Als letzte lineare dynamische Datenstruktur wird die Liste eingeführt. In dieser Sequenz liegt der Fokus auf der Möglichkeit, auf jedes Element zugreifen zu können. Die Modellierung erfolgt beim gesamten Vorhaben in Entwurfs- und Implementationsdiagrammen.

Zeitbedarf: 21 Stunden

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Die Datenstruktur Feld</p> <p>a) Erarbeitung der Anforderungen an eine Datenstruktur</p> <p>b) Wiederholung der Datenstruktur Array, Eigenschaften der Datenstruktur, Standardoperationen für ein und zwei-dimensionale Arrays</p> <p>c) Modellierung und Implementierung von Anwendungen</p>	<p>Die Schülerinnen und Schüler...</p> <ul style="list-style-type: none"> - erläutern und begründen methodische Vorgehensweisen, Entwurfs- und Implementationsentscheidungen sowie Aussagen über Informatiksysteme (A) - konstruieren zu kontextbezogenen Problemstellungen informatische Modelle (M) - ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M) 	<p>Kapitel:</p> <ul style="list-style-type: none"> - Anforderungen an eine Datenstruktur - Speichern mit Struktur <i>neue Wdh. Aufgabe entwickeln, z. B. eine Chart-Top-10, eine Aufgabe mit zweidimensionalem Array (vgl. Anforderungen KLP)</i>
<p>2. Die Datenstruktur Schlange</p> <p>a) Modellierung und Implementierung der Verknüpfung von Objekten</p> <p>b) Generische Typen, Trennung von Verwaltung und Inhalt dyn. DS.</p> <p>c) Erläuterung von Problemstellungen, die nach dem FIFO-Prinzip bearbeitet werden</p> <p>d) Funktionalität der Schlange unter Verwendung der Klasse <i>Queue</i>; Erschließen der Standardoperationen</p> <p>e) Modellierung und Implementierung einer Anwendung auf der Basis einer Anforderungsbeschreibung mit Objekten der Klasse <i>Queue</i></p>	<ul style="list-style-type: none"> - ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), - implementieren auf der Grundlage von Modellen oder Modellausschnitten Computerprogramme (I) - testen und korrigieren Computerprogramme (I) - interpretieren Fehlermeldungen und korrigieren den Quellcode (I), - überführen gegebene textuelle und grafische Darstellungen informatischer Zusammenhänge in die jeweils andere Darstellungsform (D) - stellen informatische Modelle und Abläufe in Texten, Tabellen, Diagrammen und Grafiken dar (D) - stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D) - modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M) - ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M) - dokumentieren Klassen (D) - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I) 	<p>Kapitel:</p> <ul style="list-style-type: none"> - Wer zuerst kommt - Objekte miteinander verketteten - Verwaltung und Inhalt - Funktionen der Queue <p>Aufgaben:</p> <ul style="list-style-type: none"> - Warteschlange Büro (Standardoperationen/Basiskompetenz) Kunden warten auf einem Flur, um in ein Büro vorge lassen zu werden. Sie können sich am Ende der Warteschlange anstellen, vorgelassen werden oder müssen alle gehen, wenn die Sprechzeit vorüber ist. - Erweiterte Queue Verkehrskontrolle (Vertiefung) Die Polizei kontrolliert die Fahrzeuge im Hinblick auf ihre Verkehrstauglichkeit. Für die Kontrolle werden die Fahrzeuge aus dem Verkehr gewunken. Es werden so lange Fahrzeuge kontrolliert, bis eine gewissen Menge an Verstößen vorliegt oder Autos kontrolliert wurden.
<p>3. Die Datenstruktur Stapel</p> <p>a) Erläuterung von Problemstellungen, die nach dem LIFO-Prinzip bearbeitet werden</p> <p>b) Funktionalität der Klasse Stapel unter Verwendung der Klasse <i>Stack</i></p> <p>c) Modellierung und Implementierung einer Anwendung auf Basis einer Anforderungsbeschreibung mit Objekten der Klasse <i>Queue</i></p> <p>d) Modellierung und Implementierung</p>		<p>Kapitel:</p> <ul style="list-style-type: none"> - Daten gut abgelegt – Stapel - Funktionen der Datenstruktur Stapel <p>Aufgaben:</p> <ul style="list-style-type: none"> - Standardoperationen/Basiskompetenz (Stapel Münzen/CDs) zur Umsetzung der gegebenen Funktionen der Klasse <i>Stack</i> - Biber und Teller Es gibt große und kleine Biber sowie grüne und braune

<p>ung einer Anwendung unter Verwendung verschiedener Datenstrukturen (Objekte der Klassen <i>Queue</i>, <i>Stack</i> und <i>Array</i> (<i>Palindrom</i>))</p>		<p>Teller. Es muss überprüft werden, ob die gestapelten Teller zur Schlange der Biber passen, da die großen Biber nur von den braunen Tellern essen und die kleinen von den grünen. Hierbei müssen sowohl <i>Queue</i> als auch <i>Stack</i> verwendet werden.</p> <p>- Palindrom Es wird überprüft, ob ein beliebiges Wort ein Palindrom ist.</p>
<p>4. Die Datenstruktur Liste</p> <p>a) Analyse der Möglichkeiten bisheriger Datenstrukturen zwecks Bestimmung notwendiger Funktionalitäten für komplexere Anwendungen (Abgrenzung zu <i>Stack/Queue</i>, zusätzliche Fähigkeiten der Klasse <i>List</i>)</p> <p>b) Erarbeitung der Funktionalität der Liste unter Verwendung der Klasse <i>List</i></p> <p>c) Modellierung und Implementierung einer Anwendung mit Objekten der Klasse <i>List</i></p> <p>d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (<i>Stack</i>, <i>Queue</i>, <i>List</i>)</p>		<p>Kapitel:</p> <ul style="list-style-type: none"> - Flexibel für alle Fälle – (die lineare Liste - Funktionen der Datenstruktur Liste <p>Aufgaben:</p> <ul style="list-style-type: none"> - LEDs - Textzeilen verarbeiten
<p>5. Übungen und Vertiefungen zur Verwendung linearer und dynamischer Datenstrukturen anhand weiterer Problemstellungen</p>		<p>Prüfungsvorbereitung</p>
		<p><i>Projekteinstieg Heldenspiel</i> Mit dem Heldenspiel können alle im Kapitel behandelten Datenstrukturen erarbeitet werden. Das Spiel kann bis zu einem beliebigen Grad realisiert werden, sodass es sowohl als Einstieg als auch als ein umfassendes Projekt für lineare Datenstrukturen genutzt werden kann.</p>

Unterrichtsvorhaben Q1-III

Thema: Algorithmen zum Suchen und Sortieren auf linearen Datenstrukturen

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

Vorhabenbezogene Konkretisierung:

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

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

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

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

Zeitbedarf: 21 Stunden

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

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

Unterrichtsvorhaben Q-I IV

Thema: Automaten und formale Sprachen

Leitfragen:

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

Vorhabensbezogene Konkretisierung:

Ausgehend von der Beschreibung und Untersuchung realer Automaten wird das formale Modell eines endlichen Automaten entwickelt. Neben dem Mealy-Automaten geht es vor allem um den erkennenden endlichen Automaten. Auf die Erarbeitung der Beschreibung folgt die Modellierung eigener Automaten und die Untersuchung bestehender, um die Eigenschaften und Grenzen eines endlichen Automaten zu erkennen. Hierbei wird dessen Verhalten auf bestimmte Eingaben analysiert.

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

Zeitbedarf: 21 Std.

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

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

Unterrichtsvorhaben Q2-I

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

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

Vorhabenbezogene Konkretisierung:

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

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

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

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

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

Zeitbedarf: 21 Stunden

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Aufbau von Baumstrukturen und Grundbegriffe a) Erarbeitung der Begriffe Wurzel, Knoten, Blatt, Kante, Grad eines Knotens und eines Baumes, Pfad, Tiefe, Ebene, Teilbaum b) Aufbau und Darstellung von Baumstrukturen in verschiedenen Anwendungskontexten	Die Schülerinnen und Schüler - stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D), - erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), - analysieren und erläutern Algorithmen und Programme (A), - stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D). - beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),	6.1 Spielen mit Struktur – Baumstrukturen Projekteinstieg 1: Spielbäume
2. Binäre Bäume a) rekursiver Aufbau eines binären Baums b) Traversierungen (pre-, in-, postorder) c) Modellierung eines Binärbaums in einem Anwendungskontext mit Hilfe der Klasse BinaryTree (als Entwurfs- und Implementationsdiagramm) d) Implementation einer Anwendung der Datenstruktur binärer Baum (ggf. in Teilen)	- ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), - ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), - modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), - verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M),	6.2 Zwei Nachfolger sind genug! - Binäre Bäume Implementation des Projekts Ahnenbaum
3. Binäre Suchbäume a) Prinzip des binären Suchbaums, Ordnungsrelation b) Operationen auf dem binären Suchbaum (Suchen, Einfügen, Löschen, sortierte Ausgabe) c) Modellierung eines binären Suchbaums in einem Anwendungskontext mit Hilfe der Klasse BinarySearchTree (als Entwurfs- und Implementationsdiagramm) und dem Interface Item d) Implementation einer Anwendung der Datenstruktur binärer Suchbaum (ggf. in Teilen)	- 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 den Quellcode (I), - testen Programme systematisch anhand von Beispielen (I),	6.3 Wer Ordnung hält, spart Zeit beim Suchen – Binäre Suchbäume Projekteinstieg 2: Binäre Suchbäume Implementation des Projekts Benutzerverwaltung
4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen		Prüfungsvorbereitung

Unterrichtsvorhaben Q2-II

Thema: Aufbau von und Kommunikation in Netzwerken

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

Vorhabenbezogene Konkretisierung:

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

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

Zeitbedarf: 15 Stunden

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Technische Kommunikation als Fortführung natürlicher Kommunikation a) Kommunikation im Shannon-Weaver-Modell b) Kriterien von technischen Kommunikationsarten c) Die Geschichte der technischen Kommunikation	Die Schülerinnen und Schüler - beschreiben und erläutern Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A), - nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D), - analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A).	7.1 Menschen kommunizieren – ohne und mit Technik Projekteinstieg: Kommunikation im Wilden Westen
2. Aufbau von Netzwerken und Kommunikationsregeln a) Das Netzwerk als Organisationsprinzip der Kommunikation und Möglichkeiten der Ausformung b) Geregelte technische Kommunikation durch Protokolle in Schichtenmodellen		7.2 Ohne Protokoll läuft nichts - Netzwerke
3. Aufgabenteilung in Netzwerken durch Server und Client a) Aufbau und Aufgaben der Client-Server-Struktur (b) Protokolle zwischen Client und Server		7.3 Einer für alle – Client-Server-Struktur
4. Kryptologie a) Veranschaulichen und Anwenden von symmetrischen und asymmetrischen kryptographischen Verfahren (Caesar, Vigenère, RSA) b) Bewertung der Verfahren hinsichtlich ihrer Sicherheit und ihrem Aufwand		Die digitale Welt 100 – Kryptologie
5. Übung und Vertiefung des Aufbaus von und der Kommunikation in Netzwerken		Prüfungsvorbereitung

Unterrichtsvorhaben Q2-3

Thema: Nutzung und Modellierung von relationalen Datenbanken in Anwendungskontexten

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

Vorhabenbezogene Konkretisierung:

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

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

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

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

Zeitbedarf: 21 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Nutzung von relationalen Datenbanken a) Aufbau von Datenbanksystemen und Grundbegriffe - Aufgaben und Eigenschaften eines Datenbanksystems - Erarbeitung der Begriffe Tabelle, Attribut, Attributwert, Datensatz, Datentyp, Primärschlüssel, Datenbankschema - Problematisierung von Redundanzen, Anomalien und Inkonsistenzen b) SQL-Abfragen - Erarbeitung der grundlegenden Sprachelemente von SQL (SELECT(DISTINCT), FROM, WHERE, JOIN) - Analyse und Erarbeitung von	Die Schülerinnen und Schüler - erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A), - analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A), - analysieren und erläutern eine Datenbankmodellierung (A), - erläutern die Eigenschaften normalisierter Datenbankschemata (A), - bestimmen Primär- und Sekundärschlüssel (M), - ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M), - modifizieren eine Datenbankmodellierung (M), - modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M), - überführen Datenbankschemata in vorgegebene Normalformen (M),	Kapitel 8 Datenbanken 8.1 Wissen speichern und verwalten – Datenbanksysteme 8.2 Daten anordnen mit Tabellen Beispiel: Buchhandlung Redundanzen, Anomalien und Inkonsistenzen Projekteinstieg: Tabellen 8.3 Daten filtern mit SQL 8.4 Komplexe Filter Aufgaben

<p>SQL-Abfragen (AND, OR, NOT, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, <>, >, <, >=, <=, LIKE, BETWEEN, IN, IS NULL, geschachtelte Select-Ausdrücke)</p> <p>c) Vertiefung an einem weiteren Datenbankbeispiel</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 mehrere verknüpfte Tabellen (D), - stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D), - überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D). 	
<p>2. Modellierung von relationalen Datenbanken</p> <p>a) Datenbankentwurf durch ER-Diagramme</p> <ul style="list-style-type: none"> - Ermittlung von Entitäten, zugehörigen Attributen, Beziehungen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms - Erläuterung und Erweiterung einer Datenbankmodellierung <p>b) Entwicklung eines relationalen Modells aus einem Datenbankentwurf</p> <ul style="list-style-type: none"> - Überführung eines Entity-Relationship-Diagramms in ein relationales Datenbankschema inklusive der Bestimmung von Primär- und Fremdschlüsseln <p>c) Normalformen</p> <ul style="list-style-type: none"> - Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten) 		<p>8.5 Datenbankentwurf <i>Beispiel: Online-Buchhandel Datenanalyse und Entwurf</i></p> <p>8.6 Umsetzung des ER-Modells <i>Entitätsmengen m:n-Beziehungen 1:n-Beziehungen 1:1-Beziehungen</i></p> <p><i>Wiederaufgriff des Projekteinstiegs 8.7 Datenbanken verbessern durch Normalformen</i></p>
<p>3. Übung und Vertiefung der Nutzung und Modellierung von relationalen Datenbanken</p>		<p>Prüfungsvorbereitung</p>

2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

In einer Zeit, in der Informatik immer mehr Lebensbereiche erfasst und Fachkräfte in der IT-Branche gesucht sind, brauchen Schülerinnen und Schüler zum einen fachliche Orientierung zur Einordnung der Informatik in ihrem persönlichen Umfeld, zum anderen müssen sie aber auch anschlussfähiges Wissen für eine vertiefte informatische Bildung und Ausbildung erwerben. Der beste Weg dazu liegt in frühzeitig erworbenen Kompetenzen im Fach Informatik. Der Einsatz von Computern und Informations- und Kommunikationstechnik ist ein entscheidender Bestandteil unserer gegenwärtigen und künftigen Gesellschaft. Die Vision ist, dass informatisch gebildete Menschen alle informatischen Probleme, die ihnen in ihrem Leben begegnen werden, mit Selbstvertrauen anpacken und selbstständig allein oder im Team bewältigen können. Und die Lehrenden helfen den Lernenden dabei, ihre Kompetenz zum Lösen solcher Probleme einzusetzen, zu vertiefen und auszubauen.

Ein Informatikunterricht, der dies leistet, ist anspruchsvoll.

Lehrerinnen und Lehrer haben durch den Erziehungsauftrag der allgemeinbildenden Schule die Aufgabe, professionell andere Menschen dabei zu unterstützen, sich Kenntnisse, Fähigkeiten und Bildung anzueignen, die ihnen die Kompetenz gibt, selbstständig die Herausforderungen ihres Lebens bewältigen zu können. Die Grundsätze, die die Voraussetzungen für einen guten Informatikunterricht bilden, sollen deshalb im Folgenden vorgestellt und erörtert werden.

Die Intention ist, dass alle Schülerinnen und Schüler künftig den Einsatz von Computern und Informations- und Kommunikationstechnik zu ihrem Nutzen bewältigen können. Damit soll anschlussfähiges Lernen und somit auch ein weiteres lebensbegleitendes Lernen ermöglicht werden.

Die aktive Auseinandersetzung mit den Inhalten und Methoden der Informatik im Unterricht erfordert eine intensive Nutzung der schulischen Computeranlagen. Über das Internet können Arbeitsergebnisse zwischen Schule und häuslichem Arbeitsplatz ausgetauscht werden. Die Interaktivität des Mediums Computer ermöglicht in besonderer Weise selbstständiges und selbsttätiges Arbeiten und Lernen der Schülerinnen und Schüler. Damit sind günstige Voraussetzungen zum Lernen als aktiv-konstruktiver, selbstgesteuerter, situativer und sozialer Prozess gegeben. Als Leitkonzept für die Gestaltung geeigneter Lernumgebungen eignet sich die Problemorientierung, die eine Balance zwischen Instruktion und Konstruktion einfordert. Die Themen entstammen daher möglichst realen Problemen mit Anwendungsbezug, die mehrere Sichtweisen zu einem Inhalt erlauben.

Zusammengefasst soll sich die Unterrichtsorganisation daran ausrichten, dass

- die individuelle Schülerpersönlichkeit mit ihren Vorerfahrungen, Möglichkeiten und Leistungsdispositionen im Blick ist,
- Schülerinnen und Schüler aktiv lernen,
- Schülerinnen und Schüler kooperativ lernen,
- Vorwissen abgesichert, aufgegriffen und Lernfortschritt ermöglicht wird,
- die Aufgabenstellungen grundsätzlich problemorientiert und häufig komplex sind,
- die Aufgabenstellungen auch auf Anwendung und Transfer ausgerichtet sind.

2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

Leistungsbewertung und Leistungsrückmeldung erfolgen durch die Fachlehrkräfte in Wahrnehmung ihrer pädagogischen Verantwortung auf der Grundlage des § 48 Schulgesetz (SchulG), der §§ 13 – 16 der Verordnung über den Bildungsgang und die Abiturprüfung in der gymnasialen Oberstufe (APO-GOSt) sowie des Kapitels 3 des Kernlehrplans für die Sekundarstufe II im Fach Informatik nach den von der Fachkonferenz Informatik gemäß § 70 Abs. 4 Nr. 2 Schulgesetz beschlossenen Grundsätzen, wie sie im Leistungskonzept des Vile-Gymnasiums – abrufbar unter www.ville-gymnasium.de/unterricht/allgemeines/leistungskonzept – niedergelegt sind.

1.1 Lehr- und Lernmittel

In der Oberstufe ist das Schulbuch Informatik des Schöningh-Verlages eingeführt, das themenbezogen eingesetzt wird. Desweiteren werden grafische Programmierumgebungen (Dr. Java, BlueJ, Javaeditor), Software zur Gestaltung von Klassendiagrammen, z.B. ArgoUML, und didaktische Lernumgebungen, wie z.B. Greenfoot und GLOOP.

Im Unterricht werden desweiteren selbst erstellte Arbeitsmaterialien und die Möglichkeiten des Internets mit all seinen Foren, Tutorials und Wikis genutzt.

3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen

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

Möglichst schon im zweiten Halbjahr der Einführungsphase, spätestens jedoch im ersten Halbjahr der Qualifikationsphase 1 werden im Unterricht an geeigneten Stellen Hinweise zu möglichen Themen von Facharbeiten gegeben. Im Rahmen der Methodentage werden Hinweise zu den fachspezifischen Anforderungen und zur Bewertung der Facharbeit gegeben.

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.

Außerdem werden jährlich in einer Sitzung der Fachkonferenz die Erfahrungen mit dem schulinternen Curriculum in der Unterrichtspraxis ausgetauscht und ggf. Änderungen beschlossen, um erkannten ungünstigen Entscheidungen schnellstmöglich entgegenwirken zu können. Darüber hinaus erfolgt in Kenntnis der jeweiligen Abiturvorgaben ggf. eine Anpassung der Schwerpunktsetzung für den jeweiligen Abiturjahrgang.