

St.-Anna-Schule Wuppertal

**Schulinterne Umsetzung des Kernlehrplans für die gymnasiale
Oberstufe**

Informatik

(Stand: 01.06.2016)

Inhalt

1 Die Fachgruppe Informatik der St.-Anna-Schule Wuppertal.....	1
1.1 Unterrichtsbedingungen.....	1
2 Entscheidungen zum Unterricht.....	2
2.1 Unterrichtsvorhaben.....	2
2.1.1 Übersichtsraster Unterrichtsvorhaben.....	2
2.1.2 Konkretisierte Unterrichtsvorhaben.....	6
2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit.....	36
2.3 Grundsätze der Leistungsbewertung und -rückmeldung.....	36
2.3.1 Grundsätze.....	36
2.3.2 Beurteilungsbereich Klausuren.....	36
2.3.3 Beurteilungsbereich Sonstige Mitarbeit.....	37
2.4 Lehr- und Lernmittel.....	39
3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen.....	40
4 Qualitätssicherung und Evaluation.....	41

1 Die Fachgruppe Informatik der St.-Anna-Schule Wuppertal

1.1 Unterrichtsbedingungen

Unterrichtsangebot in der Sek. 1	Klasse 8, 9 im Rahmen von WP II (Klasse 8: vierstündig; Klasse 9: zweistündig)
Unterrichtsangebot in der Sek. 2	Grundkurse (3-stündig)
Besondere unterrichtliche Angebote	Projektkurs Robotik in der Q1
Außerunterrichtliche Angebote	Robotik-AG
Sonstiges	zwei Informatikräume und ein Experimentierraum Informations- und kommunikationstechnische Grundbildung (IKG) angebunden an den Fachunterricht außerhalb des Informatikunterrichts

2 Entscheidungen zum Unterricht

Zur Einführung in die objektorientierte Programmierung wird die didaktische Lernumgebung Greenfoot gewählt.

Nach der Einarbeitungsphase wird auf eine komplexere Lernumgebung umgestellt:

- BlueJ
- Eclipse

Begleitend zur eigenen Unterrichtsplanung wird das Lehrwerk "Informatik" für die gymnasiale Oberstufe des Schöningh-Verlags verwendet.

2.1 Unterrichtsvorhaben

2.1.1 Übersichtsraster Unterrichtsvorhaben

Die in dem Übersichtsraster angegebenen zeitlichen Angaben verstehen sich als Orientierungsrahmen, der kursabhängig angemessen über- und unterschritten werden darf.

Einführungsphase

Einführungsphase – 1. Halbjahr	
<p><u>Unterrichtsvorhaben E-I</u></p> <p>Thema: Was macht Informatik? - Einführung in die Inhaltsfelder der Informatik</p> <p>Zentrale Kompetenzen:</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 E-II</u></p> <p>Thema: Grundlagen der objektorientierten Analyse, Modellierung und Implementierung</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 • 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 E-III</u></p> <p>Thema: Algorithmische Grundstrukturen in Java</p> <p>Zentrale Kompetenzen:</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 	<p><u>Unterrichtsvorhaben E-IV</u></p> <p>Thema: Das ist die digitale Welt! – Einführung in die Grundlagen, Anwendungsgebiete und Verarbeitung binärer Codierung</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Kommunizieren und Kooperieren • Darstellen und Interpretieren • Argumentieren <p>Inhaltsfelder:</p>

<ul style="list-style-type: none"> • 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>	<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: 6 Stunden</p>
--	--

Einführungsphase – 2. Halbjahr

<p><u>Unterrichtsvorhaben E-V</u></p> <p>Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand lebensnaher Anforderungsbeispiele</p> <p>Zentrale Kompetenzen:</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 • UML-Diagramme <p>Zeitbedarf: 21 Stunden</p>	<p><u>Unterrichtsvorhaben E-VI</u></p> <p>Thema: Such- und Sortieralgorithmen</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Modellieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren • Implementieren <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 • Analyse, Entwurf und Implementierung einfacher Algorithmen • Objekte und Klassen <p>Zeitbedarf: 15 Stunden</p>
<p><u>Unterrichtsvorhaben E-VII</u></p> <p>Thema: Leben in der digitalen Welt – Immer mehr Möglichkeiten und immer mehr Gefahren!?</p> <p>Zentrale Kompetenzen:</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>	
--	--

Summe Einführungsphase: ca. 81 Stunden

Qualifikationsphase 1

Qualifikationsphase 1 – 1. Halbjahr	
<p><u>Unterrichtsvorhaben Q1-I</u></p> <p>Thema: Wiederholung und Vertiefung der objektorientierten Modellierung</p> <p>Zentrale Kompetenzen:</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: 6 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>Zentrale Kompetenzen:</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: 20 Stunden</p>
<p><u>Unterrichtsvorhaben Q1-III</u></p> <p>Thema: Algorithmen zum Suchen und Sortieren auf linearen Datenstrukturen</p> <p>Zentrale Kompetenzen:</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: 20 Stunden</p>	
---	--

Qualifikationsphase 1 – 2. Halbjahr
--

<p><u>Unterrichtsvorhaben Q1-IV</u></p> <p>Thema: Automaten und formale Sprachen</p> <p>Zentrale Kompetenzen:</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: 20 Stunden</p>	<p><u>Unterrichtsvorhaben Q1-V</u></p> <p>Thema: Nutzung und Modellierung von relationalen Datenbanken in Anwendungskontexten</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten • Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Datenbanken • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache • Sicherheit <p>Zeitbedarf: 20 Stunden</p>
--	---

Summe Qualifikationsphase 1: ca. 86 Stunden
--

Qualifikationsphase 2

Qualifikationsphase 2 – 1. Halbjahr
--

<p><u>Unterrichtsvorhaben Q2-I</u></p> <p>Thema: Organisation und Verarbeitung von Daten II – Modellierung und Implementierung von Anwendungen mit dynamischen nicht-linearen Datenstrukturen</p> <p>Zentrale Kompetenzen:</p>	<p><u>Unterrichtsvorhaben Q2-II</u></p> <p>Thema: Kryptologie</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Darstellen und Interpretieren
--	---

<ul style="list-style-type: none"> • Argumentieren • Darstellen und Interpretieren • 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 • Analyse, Entwurf und Implementierung von Algorithmen • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: 36 Stunden</p>	<ul style="list-style-type: none"> • Kommunizieren und Kooperieren • Implementieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Informatiksysteme • Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Sicherheit • Nutzung von Informatiksystemen, Wirkungen der Automatisierung <p>Zeitbedarf: 16 Stunden</p>
---	--

Qualifikationsphase 2 – 2. Halbjahr
--

<p><u>Unterrichtsvorhaben Q2-III</u></p> <p>Thema: Aufbau von und Kommunikation in Netzwerken</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Informatiksysteme • Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Einzelrechner und Rechnernetzwerke • Sicherheit • Nutzung von Informatiksystemen, Wirkungen der Automatisierung <p>Zeitbedarf: 16 Stunden</p>	
---	--

Summe Qualifikationsphase 2: ca. 56 Stunden
--

2.1.2 Konkretisierte Unterrichtsvorhaben

Die folgenden Kompetenzen aus dem Bereich Kommunizieren und Kooperieren werden in allen Unterrichtsvorhaben der Einführungs- und Qualifikationsphase 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).

Einführungsphase

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.

Zum Beispiel 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 so erworbene Wissen kann auf weitere den Schülerinnen und Schülern bekannte Informatiksysteme übertragen werden.

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

Zeitbedarf: 6 Stunden

Sequenzierung des Unterrichtsvorhabens:

<u>Unterrichtssequenzen</u>	<u>Zu entwickelnde Kompetenzen</u>	<u>Kapitel und Materialien</u>
(1) Informatiksysteme und ihr genereller Aufbau <ul style="list-style-type: none"> • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten • Informatiksysteme • Informatik, Mensch und Gesellschaft 	Die Schülerinnen und Schüler ... <ul style="list-style-type: none"> • ... 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 an
(2) Der kompetente Umgang mit dem Schulnetzwerk <ul style="list-style-type: none"> • Netzwerkstruktur der Informatikräume • Einzelrechner und Netzwerk • Erstellen und Anlegen von 		Kapitel 1 "Was macht Informatik" MedienScouts, Benutzer- und Datenschutzbestimmungen der Schule

Ordnerstrukturen

- Sortieren von Dateien und Ordern
- Sicherheit und Datenschutz

--

--

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 an einem Greenfoot-Szenario 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. Der Fokus wird auf Grundlagen wie der Unterscheidung zwischen Klasse und Objekt, Attribute, Methoden, Objektidentität und Objektzustand gelegt. Die Greenfoot-Umgebung ermöglicht es, Beziehungen zwischen Klassen zu einem späteren Zeitpunkt (Kapitel 4: Methoden und Variablen) zu thematisieren.

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: Algorithmen) führen. Dort stehen Kontrollstrukturen im Mittelpunkt.

Zeitbedarf: 9 Stunden

Sequenzierung des Unterrichtsvorhabens:

<u>Unterrichtssequenzen</u>	<u>Zu entwickelnde Kompetenzen</u>	<u>Kapitel und Materialien</u>
(1) Identifikation von Objekten und Klassen <ul style="list-style-type: none">• An einem lebensweltnahen Beispiel werden Objekte und Klassen im Sinne der objektorientierten Modellierung eingeführt.• Objekte werden durch Objektdiagramme, Klassen durch Klassendiagramme dargestellt.• Die Modellierungen werden einem konkreten Anwendungsfall entsprechend angepasst.	Die Schülerinnen und Schüler ... <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	Kapitel 2 "Einführung in die Objektorientierung" 2.1 Objektorientierte Modellierung
(2) Analyse von Objekten und Klassen im Greenfoot-Szenario		Kapitel 2 "Einführung in die Objektorientierung"

<ul style="list-style-type: none"> • Schritte der objektorientierten Analyse, Modellierung und Implementation • Analyse und Erprobung der Objekte im Greenfoot-Szenario 	<p>Klassenbibliotheken (I).</p>	<p>z. B. 2.2 Das Greenfoot-Szenario „Planetenerkundung“ Von der Realität zu Objekten Von den Objekten zu Klassen, Klassendokumentation Objekte inspizieren Methoden aufrufen Objektidentität und Objektzustand</p>
<p>(3) Implementierung einfacher Aktionen in Greenfoot</p> <ul style="list-style-type: none"> • Quelltext einer Java-Klasse • Implementation eigener Methoden, Dokumentation mit JavaDoc • Programme übersetzen (Aufgabe des Compilers) und testen 		<p>Kapitel 2 "Einführung in die Objektorientierung" 2.3 Programmierung in Greenfoot Methoden schreiben 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:

<u>Unterrichtssequenzen</u>	<u>Zu entwickelnde Kompetenzen</u>	<u>Kapitel und Materialien</u>
(1) Algorithmen <ul style="list-style-type: none">• Wiederholungen (While-Schleife)• bedingte Anweisungen• Verknüpfung von Bedingungen durch die logischen Funktionen UND, ODER und NICHT• Systematisierung des Vorgehens zur Entwicklung von Algorithmen zur Lösung komplexerer Probleme		Kapitel 3 "Algorithmen" <ul style="list-style-type: none">3.1 Kontrollstrukturen3.2 Wiederholungen3.3 Zählschleifen3.4 Bedingte Anweisungen3.5 Logische Operatoren3.6 Algorithmen entwickeln
(2) Variablen und Methoden <ul style="list-style-type: none">• Implementierung eigener Methoden mit lokalen Variablen, auch zur Realisierung einer Zählschleife• Implementierung eigener Methoden mit Parameterübergabe und/oder Rückgabewert• Implementierung von Konstruktoren• Realisierung von Attributen		Kapitel 4 "Variablen und Methoden" <ul style="list-style-type: none">4.1 Variablen4.2 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: 6 Stunden

Sequenzierung des Unterrichtsvorhabens:

<u>Unterrichtssequenzen</u>	<u>Zu entwickelnde Kompetenzen</u>	<u>Kapitel und Materialien</u>
<p>(1) Analoge und digitale Aufbereitung und Verarbeitung von Daten</p> <ul style="list-style-type: none"> • Erarbeitung der Unterschiede von analog und digital • Zusammenfassung und Bewertung der technischen Möglichkeiten von analog und digital 	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> • ... 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). 	Exkurs "Analog und Digital"
<p>(2) Der Umgang mit binärer Codierung von Informationen</p> <ul style="list-style-type: none"> • Das binäre (und hexadezimale) Zahlensystem • Binäre Informationsspeicherung • Binäre Verschlüsselung • Implementation eines Binärumrechners 		Exkurs "Binäre Welt"
<p>(3) Aufbau informatischer Systeme</p> <ul style="list-style-type: none"> • Identifikation des EVA-Prinzips als grundlegende 		Exkurs "Arbeitsweise eines Computers"

Arbeitsweise informatischer
Systemen

- Nachvollziehen der von-
Neumann-Architektur als
relevantes Modell der
Umsetzung des EVA-Prinzips

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 Benutzerschnittstellen eingeführt.

Zeitbedarf: 21 Stunden

Sequenzierung des Unterrichtsvorhabens:

<u>Unterrichtssequenzen</u>	<u>Zu entwickelnde Kompetenzen</u>	<u>Kapitel und Materialien</u>
<p>(1) Umsetzung von Anforderungen in Entwurfsdiagramme</p> <ul style="list-style-type: none"> • Aus Anforderungsbeschreibungen werden Objekte mit ihren Eigenschaften identifiziert • Gleichartige Objekte werden in Klassen (Entwurf) zusammengefasst und um Datentypen und Methoden erweitert 	<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), • ... modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), • ... ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu 	<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> <ul style="list-style-type: none"> • Erweiterung des Entwurfsdiagramms um Konstruktoren und get- und set-Methoden • Festlegung von Datentypen 	This cell is shared with the previous row's list of competencies	<p>Kapitel 5 "Klassenentwurf"</p> <p>5.4 Klassen und Beziehungen implementieren</p> <p>5.5 Vererbung</p>

<p>in Java, sowie von Rückgaben und Parametern</p> <ul style="list-style-type: none"> • Entwicklung von Klassendokumentationen • Erstellung von Sequenzdiagrammen als Vorbereitung für die Programmierung 	<p>(M),</p> <ul style="list-style-type: none"> • ... 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>(3) Programmierung anhand der Dokumentation und des Implementations- und Sequenzdiagrammes</p> <ul style="list-style-type: none"> • Klassen werden in Java-Quellcode umgesetzt • Das Geheimnisprinzip wird umgesetzt • Einzelne Klassen und das Gesamtsystem werden anhand der Anforderungen und Dokumentationen auf ihre Korrektheit überprüft. 	<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), 	<p>Kapitel 5 "Klassenentwurf" 5.4 Klassen und Beziehungen implementieren 5.5 Vererbung</p>
<p>(4) Vererbungsbeziehungen</p> <ul style="list-style-type: none"> • Das Grundprinzip der Vererbung wird erarbeitet • Die Vorteile der Vererbungsbeziehungen • Vererbung wird implementiert 	<ul style="list-style-type: none"> • ... entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M), • ... stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), 	<p>Kapitel 5 "Klassenentwurf" 5.5 Vererbung</p>
<p>(5) Softwareprojekt</p> <ul style="list-style-type: none"> • Analyse und Dekonstruktion eines Spiels (Modelle, Quelltexte) • Erweiterung des Spiels um weitere Funktionalitäten • Modellierung eines Spiels aufgrund einer Anforderungsbeschreibung, inklusive einer grafischen Benutzerschnittstelle • (arbeitsteilige) Implementation des Spiels 	<ul style="list-style-type: none"> • ... dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D). 	<p>Kapitel 7 "Softwareprojekte" 7.1 Softwareentwicklung 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.

Zunächst lernen die Schülerinnen und Schüler das Feld als eine erste Datensammlung kennen. Optional können nun zunächst die wesentlichen Eigenschaften von Algorithmen wie z.B. Korrektheit, Terminiertheit, Effizienz und Verständlichkeit sowie die Schritte einer Algorithmenentwicklung erarbeitet werden (Klärung der Anforderung, Visualisierung, Zerlegung in Teilprobleme).

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: 15 Stunden

Sequenzierung des Unterrichtsvorhabens:

<u>Unterrichtssequenzen</u>	<u>Zu entwickelnde Kompetenzen</u>	<u>Kapitel und Materialien</u>
<p>(1) Modellierung und Implementation von Datenansammlungen</p> <ul style="list-style-type: none"> • Modellierung von Attributen als Felder • Deklaration, Instanziierung und Zugriffe auf ein Feld 	<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> <ul style="list-style-type: none"> • Erkundung von Strategien für das Suchen auf unsortierten Daten, auf sortierten Daten und mithilfe einer Berechnungsfunktion. • Vergleich der drei Verfahren durch intuitive Effizienzbetrachtungen. 		<p>Kapitel 6 "Suchen und Sortieren"</p> <p>Projekteinstieg 1: Suchen</p> <p>6.2 Suchen mit System, z. B.: Lineare Suche Binäre Suche Hashing</p>
<p>(3) Systematisierung von Algorithmen und Effizienzbetrachtungen</p> <ul style="list-style-type: none"> • Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode • Anwendung von Sortieralgorithmen auf 		<p>Kapitel 6 "Suchen und Sortieren"</p> <p>Projekteinstieg 2: Sortieren, z. B.: 6.3 Ordnung ist das halbe Leben!? – Sortieren</p> <p>Sortierverfahren: Selection Sort, Insertion Sort, Bubble Sort</p>

verschiedene Beispiele

- Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche
- Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des Speicherplatzbedarfs
- Analyse eines weiteren Sortieralgorithmus (sofern nicht in (a) bereits geschehen)

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 werden 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:

<u>Unterrichtssequenzen</u>	<u>Zu entwickelnde Kompetenzen</u>	<u>Kapitel und Materialien</u>
<p>(1) Schriftzeichen, Rechenmaschine, Computer</p> <ul style="list-style-type: none"> • Anhand von Schwerpunkten, wie z.B. Datenspeicherung, Maschinen, Vernetzung sollen wichtige Entwicklungen der Informatik vorgestellt werden. • Anhand der unterschiedlichen Schwerpunkte sollen universelle Tendenzen der Entwicklung der Informationsverarbeitung erarbeitet werden. 	<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), • ... nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K). 	<p>Exkurs "Geschichte der Informatik"</p>
<p>(2) Die Informationsverarbeitung und ihre Möglichkeiten und Gefahren</p> <ul style="list-style-type: none"> • Ausgehend von 1. werden Tendenzen der Entwicklung der Informatik erarbeitet • Informatik wird als Hilfswissenschaft klassifiziert, die weit über ihren originären Bereich hinaus Effizienz- und Leistungssteigerungen erzeugt 		<p>Exkurs "Informatik und Gesellschaft"</p>

<ul style="list-style-type: none"> Anhand von Fallbeispielen werden technische und organisatorische Vorteile, sowie deren datenschutzrechtlichen Nachteile betrachtet. 		
---	--	--

Qualifikationsphase 1

Unterrichtsvorhaben Q1-I		
Thema: Wiederholung und Vertiefung der objektorientierten Modellierung		
Leitfragen:		
<ul style="list-style-type: none"> 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:		
<p>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.</p>		
Zeitbedarf: 6 Stunden		
Sequenzierung des Unterrichtsvorhabens:		
<u>Unterrichtssequenzen</u>	<u>Zu entwickelnde Kompetenzen</u>	<u>Kapitel und Materialien</u>
(1) Wiederholung der grundlegenden Konzepte der objektorientierten Programmierung <ul style="list-style-type: none"> Sichtweise der objektorientierten Informatik auf die Welt OOP als informatikspezifische Modellierung der Realität Schritte der Softwareentwicklung 	Die Schülerinnen und Schüler ... <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 	Kapitel 1 "Konzepte des objektorientierten Modellierens" 1.1 Modellierung der Realität 1.2 Die Welt ist voller Objekte Projekteinstieg: Klassenentwurf – step by step

<p>(2) Erweiterung der objektorientierten Programmierung</p> <ul style="list-style-type: none"> • Umsetzung einer Anforderung in Entwurfs- und Klassendiagramm • Objektkommunikation im Sequenzdiagramm • Klassendokumentation • Umsetzung von Teilen der Modellierung 	<p>Sichtbarkeitsbereiche zu (M),</p> <ul style="list-style-type: none"> • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • wenden eine didaktisch orientierte 	<p>Kapitel 1 "Konzepte des objektorientierten Modellierens" 1.3 Gut geplant – Klassenentwurf 1.4 Hierarchien machen es einfacher – Vererbung</p>
<p>(3) Mensch und Technik</p> <ul style="list-style-type: none"> • Verantwortung von Informatikern • Automatisierung des Alltags durch Informatik 	<p>Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I),</p>	<p>Kapitel 1 "Konzepte des objektorientierten Modellierens" Die digitale Welt 001 – Mensch und Technik</p>
<p>(4) Übung und Vertiefung der OOM / OOP</p>	<ul style="list-style-type: none"> • stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen (D), • 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), • 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>Kapitel 1 "Konzepte des objektorientierten Modellierens"</p>

Unterrichtsvorhaben Q1-II

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?

Vorhabenbezogene 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. Nachdem die umfangreicheren Standardoperationen dieser Datenstruktur in einem einführenden Beispiel (Vokabeltrainer) erarbeitet und in einem weiteren Beispiel vertieft (LED) wurden, werden abschließend in einem Anwendungskontext verschiedene lineare Datenstrukturen angewendet. Die Modellierung erfolgt beim gesamten Vorhaben in Entwurfs- und Implementationsdiagrammen.

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

<u>Unterrichtssequenzen</u>	<u>Zu entwickelnde Kompetenzen</u>	<u>Kapitel und Materialien</u>
(1) Die Datenstruktur Feld <ul style="list-style-type: none">• Erarbeitung der Anforderungen an eine Datenstruktur• Wiederholung der Datenstruktur Array, Eigenschaften der Datenstruktur, Standardoperationen für ein- und zweidimensionale Arrays• Modellierung und Implementierung von Anwendungen	<ul style="list-style-type: none">• Die Schülerinnen und Schüler ...• erläutern und begründen methodische Vorgehensweisen, Entwurfs- und Implementationsentscheidungen sowie Aussagen über Informatiksysteme (A),• konstruieren zu kontextbezogenen Problemstellungen informatische Modelle (M),	Kapitel 2 "Lineare Datenstrukturen" 2.1 Anforderungen an eine Datenstruktur Speichern mit Struktur: <ul style="list-style-type: none">• neue Wdh. Aufgabe entwickeln, z. B. eine Chart-Top-10,• eine Aufgabe mit zweidimensionalem Array (vgl. Anforderungen KLP)

<p>(2) Die Datenstruktur Schlange</p> <ul style="list-style-type: none"> • Modellierung und Implementierung der Verknüpfung von Objekten • Generische Typen, Trennung von Verwaltung und Inhalt dyn. DS. • Erläuterung von Problemstellungen, die nach dem FIFO-Prinzip bearbeitet werden • Funktionalität der Schlange unter Verwendung der Klasse Queue; Erschließen der Standardoperationen • Modellierung und Implementierung einer Anwendung auf der Basis einer Anforderungsbeschreibung mit Objekten der Klasse Queue 	<ul style="list-style-type: none"> • 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), • 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 2 "Lineare Datenstrukturen"</p> <p>2.2 Datensammlungen fester Größe – Arrays</p> <p>2.3 Wer zuerst kommt, ... - Schlangen</p> <p>Objekte miteinander verketteten Verwaltung und Inhalt Funktionen der Queue Aufgaben, z. B.:</p> <ul style="list-style-type: none"> • Warteschlange Büro (Standardoperationen/Basiskompetenz): Kunden warten auf einem Flur, um in ein Büro vorgelassen 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 gewisse Menge an Verstößen vorliegt oder Autos kontrolliert wurden.
<p>(3) Die Datenstruktur Stapel</p> <ul style="list-style-type: none"> • Erläuterung von Problemstellungen, die nach dem LIFO-Prinzip bearbeitet werden • Funktionalität der Klasse Stapel unter Verwendung der Klasse Stack • Modellierung und Implementierung einer Anwendung auf Basis einer Anforderungsbeschreibung mit Objekten der Klasse Queue • Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (Objekte der Klassen Queue, Stack und Array (Palindrom)) 		<p>Kapitel 2 "Lineare Datenstrukturen"</p> <p>2.4 Daten gut abgelegt – Stapel</p> <p>Funktionen der Datenstruktur Stapel</p> <p>Aufgaben, z. B.:</p> <ul style="list-style-type: none"> • Standardoperationen/Basiskompetenz (Stapel Münzen/CDs) zur Umsetzung der gegebenen Funktionen der Klasse Stack • Biber und Teller: Es gibt große und kleine Biber sowie grüne und braune 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 Queue als auch Stack verwendet werden. • Palindrom: Es wird überprüft, ob ein beliebiges Wort ein Palindrom ist.
<p>(4) Die Datenstruktur Liste</p>		<p>Kapitel 2 "Lineare</p>

<ul style="list-style-type: none"> • Analyse der Möglichkeiten bisheriger Datenstrukturen zwecks Bestimmung notwendiger Funktionalitäten für komplexere Anwendungen (Abgrenzung zu Stack/Queue, zusätzliche Fähigkeiten der Klasse List) • Erarbeitung der Funktionalität der Liste unter Verwendung der Klasse List • Modellierung und Implementierung einer Anwendung mit Objekten der Klasse List • Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (Stack, Queue, List) 		<p>Datenstrukturen"</p> <p>2.5 Flexibel für alle Fälle – lineare Listen</p> <p>Funktionen der Datenstruktur Liste</p> <p>Aufgaben, z. B.:</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>Kapitel 2 "Lineare Datenstrukturen"</p> <p>2.6 Prüfungsvorbereitung</p> <p>Projekteinstieg Heldenspiel: 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 Algorithmuseigenschaften (wie Korrektheit, Effizienz und Verständlichkeit) erarbeitet und Schritte der Algorithmusentwicklung 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: 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

<u>Unterrichtssequenzen</u>	<u>Zu entwickelnde Kompetenzen</u>	<u>Kapitel und Materialien</u>
<p>(1) Eigenschaften von Algorithmen</p> <ul style="list-style-type: none"> • Qualitätseigenschaften von Algorithmen • Strukturierung von Algorithmen mit Hilfe der Strategien „Modularisierung“ und „Teile und Herrsche“ • Analyse und Entwicklung von rekursiven Algorithmen 	<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), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • testen Programme systematisch anhand von Beispielen (I). 	<p>Kapitel 3 "Algorithmen"</p> <p>3.1 Ohne Algorithmen läuft nichts 3.2 Teile die Arbeit – rekursive Algorithmen</p>
<p>(2) Suchen in Listen und Arrays</p> <ul style="list-style-type: none"> • Lineare Suche in Listen und Arrays • Binäre Suche in einem Array • Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf 		<p>Kapitel 3 "Algorithmen"</p> <p>3.3 Suchen – iterativ und rekursiv</p>
<p>(3) Sortieren auf Listen und Arrays</p>		<p>Kapitel 3 "Algorithmen"</p> <p>3.4 Sortieren – iterativ und rekursiv</p>

<ul style="list-style-type: none"> • Entwicklung und Implementierung eines iterativen Sortierverfahrens für eine Liste • Entwicklung und Implementierung eines rekursiven Sortierverfahrens für ein Array • Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf 	<ul style="list-style-type: none"> • 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), • 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). 	
---	---	--

Unterrichtsvorhaben Q1-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?

Vorhabenbezogene 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: 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

<u>Unterrichtssequenzen</u>	<u>Zu entwickelnde Kompetenzen</u>	<u>Kapitel und Materialien</u>
<p>(1) Endliche Automaten</p> <ul style="list-style-type: none">• Erarbeitung der formalen Beschreibung eines Mealy-Automaten und der Darstellungsformen• Erarbeitung der formalen Beschreibung eines deterministischen endlichen Automaten (DEA) sowie dessen Darstellungsformen; Erschließung der Fachbegriffe Alphabet, Wort, (akzeptierte) Sprache, Determinismus• Analyse der Eigenschaften von DEA durch die Modellierung eines Automaten zu einer gegebenen Problemstellung,	<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	<p>Kapitel 4 "Endliche Automaten und Formale Sprachen"</p> <p>4.1 Endliche Automaten</p> <p>Vom realen Automaten zum Modell</p> <p>Projekteinstieg: Treasure Hunt</p> <p>Der Mealy-Automat</p> <p>Der erkennende endliche Automat</p> <p>Wort und Sprache</p>

<p>der Modifikation eines Automaten sowie die Überführung der gegebenen Darstellungsform in eine andere</p>	<p>überführen sie in die jeweils andere Darstellungsform (D),</p> <ul style="list-style-type: none"> entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M), 	
<p>(2) Grammatiken regulärer Sprachen</p> <ul style="list-style-type: none"> Erarbeitung der formalen Beschreibung einer regulären Grammatik (Sprache, Terminal und Nicht-Terminal, Produktionen und Produktionsvorschriften) Analyse der Eigenschaften einer regulären Grammatik durch deren Entwicklung und Modellierung zu einer gegebenen Problemstellung. 	<ul style="list-style-type: none"> analysieren und erläutern Grammatiken regulärer Sprachen (A), 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), 	<p>Kapitel 4 "Endliche Automaten und Formale Sprachen" 4.2 Formale Sprachen Grammatiken regulärer Sprachen</p>
<p>(3) Übungen und Vertiefungen</p> <ul style="list-style-type: none"> Verwendung endlicher Automaten und Grammatiken regulärer Sprachen 	<ul style="list-style-type: none"> zeigen die Grenzen endlicher Automaten und regulärer Grammatiken auf. 	<p>Kapitel 4 "Endliche Automaten und Formale Sprachen" 4.3 Vom Text zum Programm 4.4 Scanner, Parser, Interpreter 4.5 Prüfungsvorbereitung Projekteinstieg: Erarbeitung der formalen Beschreibung und Überprüfung des Verhaltens eines erkennenden Automaten auf bestimmte Eingaben</p>

Unterrichtsvorhaben Q1-V

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: 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

<u>Unterrichtssequenzen</u>	<u>Zu entwickelnde Kompetenzen</u>	<u>Kapitel und Materialien</u>
<p>(1) Nutzung von relationalen Datenbanken</p> <ul style="list-style-type: none"> • 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 • SQL-Abfragen • Erarbeitung der grundlegenden Sprachelemente von SQL (SELECT(DISTINCT), FROM, WHERE, JOIN) • Analyse und Erarbeitung von 	<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), 	<p>Kapitel 7 Datenbanken</p> <p>7.1 Wissen speichern und verwalten – Datenbanksysteme</p> <p>7.2 Daten anordnen mit Tabellen</p> <p>Beispiel: Buchhandlung</p> <p>Redundanzen, Anomalien und Inkonsistenzen</p> <p>Projekteinstieg: Tabellen</p> <p>7.3 Operationen auf Tabellen</p> <p>7.4 Daten filtern mit SQL</p> <p>Aufgaben</p> <p>Lernumgebung auf: http://www.imoodle.de/sqltutorial/index.html</p>

<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> <ul style="list-style-type: none"> • Vertiefung an einem weiteren Datenbankbeispiel 	<ul style="list-style-type: none"> • modifizieren eine Datenbankmodellierung (M), • modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M), • überführen Datenbankschemata in vorgegebene Normalformen (M), • verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I), • ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D), • stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D), • überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D). 	
<p>(2) Modellierung von relationalen Datenbanken</p> <ul style="list-style-type: none"> • Datenbankentwurf durch ER-Diagramme <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 • Entwicklung eines relationalen Modells aus einem Datenbankentwurf <ul style="list-style-type: none"> • Überführung eines Entity-Relationship-Diagramms in ein relationales Datenbankschema inklusive der Bestimmung von Primär- und Fremdschlüsseln • Normalformen <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>Kapitel 7 Datenbanken 7.5 Datenbankentwurf Beispiel: Online-Buchhandel Datenanalyse und Entwurf 7.6 Umsetzung des ER-Modells Entitätsmengen</p> <ul style="list-style-type: none"> • m:n-Beziehungen • 1:n-Beziehungen • 1:1-Beziehungen <p>Wiederaufgriff des Projekteinstiegs 7.7 Datenbanken verbessern durch Normalformen</p>
<p>(3) Übung und Vertiefung der Nutzung und Modellierung von relationalen Datenbanken</p>		<p>Kapitel 7 Datenbanken 7.8 Prüfungsvorbereitung</p>

Qualifikationsphase 2

Unterrichtsvorhaben Q2-I		
Thema: Organisation und Verarbeitung von Daten II – Modellierung und Implementierung von Anwendungen mit dynamischen nicht-linearen Datenstrukturen		
Leitfragen:		
<ul style="list-style-type: none"> • 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:		
<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>		
Zeitbedarf: 36 Stunden		
Sequenzierung des Unterrichtsvorhabens:		
<u>Unterrichtssequenzen</u>	<u>Zu entwickelnde Kompetenzen</u>	<u>Kapitel und Materialien</u>
<p>(1) Aufbau von Baumstrukturen und Grundbegriffe</p> <ul style="list-style-type: none"> • Erarbeitung der Begriffe Wurzel, Knoten, Blatt, Kante, Grad eines Knotens und eines Baumes, Pfad, Tiefe, Ebene, Teilbaum • Aufbau und Darstellung von Baumstrukturen in verschiedenen Anwendungskontexten 	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> • 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 	<p>Kapitel 5 "Nichtlineare Datenstrukturen"</p> <p>5.1 Spielen mit Struktur – Baumstrukturen</p> <p>Projekteinstieg 1: Spielbäume</p>
<p>(2) Binäre Bäume</p>		<p>Kapitel 5 "Nichtlineare</p>

<ul style="list-style-type: none"> • rekursiver Aufbau eines binären Baums • Traversierungen (pre-, in-, postorder) • Modellierung eines Binärbaums in einem Anwendungskontext mit Hilfe der Klasse BinaryTree (als Entwurfs- und Implementationsdiagramm) • Implementation einer Anwendung der Datenstruktur binärer Baum (ggf. in Teilen) 	<ul style="list-style-type: none"> grafisch dar (D), • 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), 	<p>Datenstrukturen"</p> <p>5.2 Zwei Nachfolger sind genug! – Binäre Bäume</p> <p>Aufgaben, z. B.:</p> <ul style="list-style-type: none"> • Implementation des Projekts Ahnenbaum
<p>(3) Binäre Suchbäume</p> <ul style="list-style-type: none"> • Prinzip des binären Suchbaums, Ordnungsrelation • Operationen auf dem binären Suchbaum (Suchen, Einfügen, Löschen, sortierte Ausgabe) • Modellierung eines binären Suchbaums in einem Anwendungskontext mit Hilfe der Klasse BinarySearchTree (als Entwurfs- und Implementationsdiagramm) und dem Interface Item • Implementation einer Anwendung der Datenstruktur binärer Suchbaum (ggf. in Teilen) 	<ul style="list-style-type: none"> • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), 	<p>Kapitel 5 "Nichtlineare Datenstrukturen"</p> <p>5.3 Wer Ordnung hält, spart Zeit beim Suchen – Binäre Suchbäume</p> <p>Projekteinstieg 2: Binäre Suchbäume</p> <p>Aufgaben, z. B.:</p> <ul style="list-style-type: none"> • Implementation des Projekts Benutzerverwaltung
<p>(4) Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen</p>	<ul style="list-style-type: none"> • 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). 	<p>Kapitel 5 "Nichtlineare Datenstrukturen"</p> <p>5.4 Navigieren mit Struktur – Graphen</p> <p>5.5 Modellierung von Graphen</p> <p>5.6 Kürzester Weg – Der Algorithmus von Dijkstra</p> <p>5.7 Graphen durchsuchen</p> <p>5.8 Prüfungsvorbereitung</p>

Unterrichtsvorhaben Q2-II

Thema: Kryptologie

Leitfragen:

- Wie geheim ist geheim?
- Wie kann sicher kommuniziert werden?

Vorhabenbezogene Konkretisierung:

Sichere Kommunikation ist nur dank kryptografischer Verfahren möglich. Ausgehend von einer historischen Betrachtung werden symmetrische und asymmetrische Verfahren erläutert, angewandt und bewertet.

Zeitbedarf: 16 Stunden

Sequenzierung des Unterrichtsvorhabens:

<u>Unterrichtssequenzen</u>	<u>Zu entwickelnde Kompetenzen</u>	<u>Kapitel und Materialien</u>
(1) Spionage-Camp <ul style="list-style-type: none">• Codierung• Stenographie• Transposition• Substitution• Schlüsselaustausch	Die Schülerinnen und Schüler ... <ul style="list-style-type: none">• erarbeiten sich die (historischen) Grundlagen der Kryptologie (Y)• analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A).	Lernumgebung/ Spionage-Koffer
(2) Kryptologie <ul style="list-style-type: none">• Veranschaulichen und Anwenden von symmetrischen und asymmetrischen kryptographischen Verfahren (Caesar, Vigenère, RSA)• Bewertung der Verfahren hinsichtlich ihrer Sicherheit und ihrem Aufwand		Kapitel 6 "Kommunikation in Netzwerken" Die digitale Welt 111 – Kryptologie

Unterrichtsvorhaben Q2-III

Thema: Aufbau von und Kommunikation in Netzwerken

Leitfragen:

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

Vorhabenbezogene Konkretisierung:

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: 16 Stunden

Sequenzierung des Unterrichtsvorhabens:

<u>Unterrichtssequenzen</u>	<u>Zu entwickelnde Kompetenzen</u>	<u>Kapitel und Materialien</u>
<p>(1) Technische Kommunikation als Fortführung natürlicher Kommunikation</p> <ul style="list-style-type: none"> • Kommunikation im Shannon-Weaver-Modell • Kriterien von technischen Kommunikationsarten • Die Geschichte der technischen Kommunikation 	<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), • - 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). 	<p>Kapitel 6 "Kommunikation in Netzwerken"</p> <p>6.1 Menschen kommunizieren – ohne und mit Technik</p> <p>Projekteinstieg: Kommunikation im Wilden Westen</p>
<p>(2) Aufbau von Netzwerken und Kommunikationsregeln</p> <ul style="list-style-type: none"> • Das Netzwerk als Organisationsprinzip der Kommunikation und Möglichkeiten der Ausformung • Geregeltete technische Kommunikation durch Protokolle in Schichtenmodellen 		<p>Kapitel 6 "Kommunikation in Netzwerken"</p> <p>6.2 Ohne Protokoll läuft nichts – Netzwerke</p>
<p>(3) Aufgabenteilung in Netzwerken durch Server und Client</p>		<p>Kapitel 6 "Kommunikation in Netzwerken"</p>

<ul style="list-style-type: none"> • Aufbau und Aufgaben der Client-Server-Struktur • Protokolle zwischen Client und Server 		6.3 Einer für alle – Client-Server-Netzwerke
(4) Übung und Vertiefung des Aufbaus von und der Kommunikation in Netzwerken		Kapitel 6 "Kommunikation in Netzwerken" 6.4 Prüfungsvorbereitung

Am Ende der Qualifikationsphase soll die verbleibende Unterrichtszeit zur vertiefenden Wiederholung und Vorbereitung zum Abitur genutzt werden.

2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

- Der Unterricht ist problemorientiert und soll von realen Problemen ausgehen und sich auf solche rückbeziehen.
- Der Unterricht folgt dem Prinzip der Exemplarizität und soll ermöglichen, informatische Strukturen und Gesetzmäßigkeiten in den ausgewählten Problemen und Projekten zu erkennen.
- Der Unterricht ist anschaulich sowie gegenwarts- und zukunftsorientiert und gewinnt dadurch für die Schülerinnen und Schüler an Bedeutsamkeit.
- Der Unterricht ist projektorientiert.
- Im Unterricht werden sowohl für die Schule didaktisch reduzierte als auch reale Informatiksysteme aus der Wissenschafts-, Berufs- und Lebenswelt eingesetzt.
- Der Unterricht beinhaltet reale Begegnung mit Informatiksystemen.

2.3 Grundsätze der Leistungsbewertung und -rückmeldung

2.3.1 Grundsätze

Die Grundsätze zur Leistungsbewertung ergeben sich aus den entsprechenden Bestimmungen des Schulgesetzes bzw. des Erzbischöflichen Schulgesetzes. Für das Verfahren der Leistungsbewertung gelten die entsprechenden Paragraphen der Ausbildungs- und Prüfungsordnungen APO-SI und APO-GOST.

Die Leistungsbewertung soll Orientierung bieten und ist damit die Grundlage zur weiteren Förderung und Beratung der Schülerinnen und Schüler, zu Schullaufbahnentscheidungen, sowie zur Beratung der Erziehungsberechtigten.

Eine verantwortliche Leistungsmessung trägt außerdem zur Orientierung der Lehrerinnen und Lehrer bei, um ihren Unterricht ausgerichtet an Stärken und Schwächen der Schülerinnen und Schüler weiter zu planen und geeignete Maßnahmen zur Förderung zu finden.

Leistungsbewertung ist ein kontinuierlicher Prozess und bewertet alle von Schülerinnen und Schülern im Zusammenhang mit Unterricht erbrachten Leistungen. Sie bezieht sich auf die im Unterricht vermittelten Kenntnisse und Fertigkeiten. Das setzt voraus, dass die Schülerinnen und Schüler im Unterricht die Gelegenheit hatten, die entsprechenden Anforderungen in Umfang und Anspruch kennen zu lernen und einzuüben.

Der Lehrer muss den Schülerinnen und Schülern hinreichend Gelegenheit geben, die geforderten Leistungen auch zu erbringen, d.h. ihnen gegebenenfalls die mögliche Hilfestellung bieten, damit alle Schülerinnen und Schüler Leistung auch erbringen können. Die Bewertung der Leistungen muss den Schülerinnen und Schülern auch im Vergleich zu den Mitschülern transparent sein und mit den Schülerinnen und Schülern zum Halbjahresbeginn besprochen werden.

2.3.2 Beurteilungsbereich Klausuren

Die Klausurtermine werden jeweils zu Anfang des Halbjahres festgelegt und über das schulinterne Kommunikationssystem bekannt gemacht. Pro Halbjahr wird i. A. ein Nachschreibtermin zentral angesetzt.

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.

Anzahl, Dauer, Formate der Klausuren:

- Einführungsphase: 1 Klausur je Halbjahr 2 Unterrichtsstunden

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

Die Aufgabentypen, sowie die Anforderungsbereiche I-III sind entsprechend den Vorgaben in Kapitel 3 des Kernlehrplans zu beachten. Die Schülerinnen und Schüler werden im Unterricht mit den Aufgabenformaten des Zentralabiturs vertraut gemacht. Die Bearbeitung der Klausuraufgaben erfolgt ohne Nutzung von Informatiksystemen.

Korrektur und Bewertung der schriftlichen Leistungen

Die Korrektur von Klassenarbeiten und Klausuren (bzw. Facharbeiten) zeigt die Mängel, Fehler und Vorzüge der Arbeit auf und führt zu einer transparenten Bewertung. Die vorgegebenen Korrektur- und Fehlerzeichen sind zu beachten.

Die Korrektur einer Klausur mündet in einer nachvollziehbaren Begründung der Note, die die Vorzüge und Mängel der Arbeit deutlich und somit die Beurteilung transparent macht. Die Verwendung eines Punkterasters sichert die Transparenz und kann eine ausführliche verbalisierte Notenbegründung ersetzen.

Die Beurteilung erfolgt in der Einführungsphase in ganzen Noten. Notentendenzen haben informativen Charakter. In der Qualifikationsphase werden Teilnoten entsprechend der Notenpunkte vergeben. 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.

Bei der Bewertung der Facharbeit werden inhaltlicher Ertrag, formale Gestaltung und Eigenständigkeit der Arbeit in angemessenem Umfang berücksichtigt. Die Notengebung wird in einem kurzen abschließenden Gutachten gegeben.

2.3.3 Beurteilungsbereich Sonstige Mitarbeit

Formen der sonstigen Mitarbeit

Zu Formen der sonstigen Mitarbeit gehören alle Beiträge, die nicht Klausuren oder Facharbeiten sind:

- Qualität und Quantität der mündlichen Mitarbeit,
- nachgewiesene Konzentration bzw. Aufmerksamkeit im Unterricht,
- mündlichen oder schriftlichen Überprüfungen,
- Mitarbeit bei Einzel-, Partner- und Gruppenarbeiten,
- Präsentation von Arbeitsergebnissen,
- Präsentation der Hausaufgaben,
- Protokolle,
- Referate,
- Software-Projekte: Eigenständigkeit, Modellierung, Implementierung, Test, Anwendung, Dokumentation und Präsentation.

Der Fachlehrer teilt im Sinne einer Beurteilungstransparenz den Lerngruppen zu Beginn

des Halbjahres verbindlich mit, welche Aspekte der vielfältigen Formen der sonstigen Mitarbeit von ihm zur Leistungsbewertung herangezogen werden.

Bewertung der sonstigen Mitarbeit

Grundlegende Kriterien für die Bewertung der mündlichen Leistung sind Qualität und Kontinuität; diese haben Vorrang vor der Quantität der Beiträge. Auch die angemessene sprachliche Darstellung ist zu berücksichtigen.

Eine Orientierung für die Bewertung mündlicher Leistungen bietet die folgende Tabelle:

Leistungsbeschreibung	Note
<p>a permanente Mitarbeit im Unterricht</p> <p>b sprachlich präzise, differenzierte und argumentativ stimmige Beiträge mit sicherer Anwendung der Fachsprache; produktive und gesprächsfördernde Beiträge mit einem hohen Maß an Selbstständigkeit, kritischem Denken und Problembewusstsein; vollständig selbstgesteuerte Projektbearbeitung; Teamfähigkeit</p> <p>c fundierte Kenntnisse der Unterrichtsinhalte und sachgerechtes Einbringen</p> <p>d sehr hohe Aufmerksamkeit auf das Unterrichtsgeschehen und die Beiträge der übrigen Teilnehmer/innen</p>	sehr gut
<p>a regelmäßige und häufige Mitarbeit im Unterricht</p> <p>b sprachlich präzise und im Wesentlichen angemessene und differenzierte Beiträge, auch gelegentlich spontan, i.d.R. sichere Anwendung der Fachsprache; überwiegend eigenständige Beiträge, Aufnehmen und Verwerten von Impulsen, i.d.R. selbstständiges Schlussfolgern und Urteilen; weitgehend selbstgesteuerte Projektbearbeitung; Teamfähigkeit</p> <p>c im Wesentlichen fundierte Kenntnisse der Unterrichtsinhalte</p> <p>d hohe Aufmerksamkeit auf das Unterrichtsgeschehen und die Beiträge der anderen Teilnehmer/innen</p>	gut
<p>a häufige, aber keine durchgängige Mitarbeit</p> <p>b zusammenhängend bzw. flüssig formulierte Beiträge, sachgerechte Formulierungen, nicht durchgängige Anwendung der Fachsprache; Beiträge meist rezeptiv, gelegentlich produktiv; Zusammenhänge werden erkannt, ansatzweise Transferleistung, aber auf Lenkung angewiesen; impulsgesteuerte Projektbearbeitung; Teamfähigkeit</p> <p>c Kenntnisse können gezielt wiedergegeben werden</p> <p>d im Wesentlichen aufmerksame Teilnahme am Unterrichtsgeschehen</p>	befriedigend
<p>a punktuelle Mitarbeit</p> <p>b grundlegende Kenntnisse des Fachvokabulars; weitgehend reproduktive Beiträge mit geringem inhaltlichem Ertrag, bei Nachfragen ist Mitdenken erkennbar; angeleitete Projektbearbeitung</p> <p>c grundlegende Kenntnisse können auf Ansprache wiedergegeben werden</p> <p>d passive Aufmerksamkeit</p>	ausreichend
<p>a vereinzelte Äußerungen</p> <p>b inhaltlich wenig ergiebige, keine selbst initiierte Mitarbeit; auch auf Ansprache</p>	mangelhaft

Hardware

- MacBook Air 11
- LEGO Mindstorms

3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen

Im ersten Jahr der Qualifikationsphase wird der Projektkurs "Robotik" in Kooperation mit der Bergischen Universität zu Wuppertal angeboten. Das Referenzfach ist Informatik.

Im Rahmen einer besonderen Lernleistung kann in Zusammenarbeit mit der Barmenia-Versicherung ein Projekt im IT-Bereich des Unternehmens umgesetzt werden.

Zusätzlich können im Bereich der schulinternen EDV Projektarbeiten vergeben werden.

4 Qualitätssicherung und Evaluation

Evaluation des schulinternen Curriculums

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

Prozess: Der Prüfmodus erfolgt jährlich. Es werden die Erfahrungen des vergangenen Schuljahres in der Fachschaft gesammelt, bewertet und eventuell notwendige Konsequenzen formuliert. Der im Anhang befindliche Evaluationsbogen kann als Instrument einer solchen Bilanzierung genutzt werden. Gegebenenfalls werden Änderungen des schulinternen Curriculums auf der Fachkonferenz beraten und vorgenommen.

Regelmäßig pflegt die Fachgruppe Informatik einen Materialaustausch über die schulinterne Lernumgebung moodle und steht im regelmäßigen Austausch über die konkrete unterrichtliche Umsetzung der Lehrinhalte und deren Überprüfungen. Dadurch wird ein hohes Maß an fachlicher Qualitätssicherung erreicht.