

Schulinternen Lehrplan

zum Kernlehrplan für die gymnasiale Oberstufe

Informatik

(Stand: 26.11.2014)

Inhalt

1	Die Fachgruppe Informatik des Lise-Meitner-Gymnasiums Geldern	3
2	Entscheidungen zum Unterricht	4
	2.1 Unterrichtsvorhaben	4
	2.1.1 Übersichtsraster Unterrichtsvorhaben	6
	2.1.2 Konkretisierte Unterrichtsvorhaben	11
	2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit	34
	2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung	35
	2.3.1 Beurteilungsbereich Klausuren	35
	2.3.2 Beurteilungsbereich Sonstige Mitarbeit	36
3	Entscheidungen zu fach- und unterrichtsübergreifenden Fragen	37
4	Qualitätssicherung und Evaluation	38

1 Die Fachgruppe Informatik des Lise-Meitner-Gymnasiums Geldern

Das Lise-Meitner-Gymnasium ist eine vierzügige Schule am Stadtrand von Geldern mit zurzeit ca. 1000 Schülerinnen und Schülern und 70 Lehrerinnen und Lehrern. Das Einzugsgebiet der Schule umfasst neben der Stadt Geldern die benachbarte Ortschaften Kapellen, Alpen und Sonsbeck. Im Bereich der Sekundarstufe II kooperiert das Gymnasium mit seiner Nachbarschule, dem Friedrich-Spee-Gymnasium und bietet mit ihm zahlreiche gemeinsame Kurse an, unter anderem in Informatik.

Am Lise-Meitner-Gymnasium wird Informatik ab der Jahrgangsstufe 8 im Wahlpflichtbereich II (WP II) zweistündig unterrichtet, wobei jedes Jahr in der Regel ein bis zwei Kurse angeboten werden können. In der zweijährigen Laufzeit dieser Kurse wird in altersstufengerechter Weise unter anderem auf Grundlagen der Algorithmik am Beispiel einer didaktischen Lernumgebung wie z.B. Kara , auf die technische Informatik und auf Robotik eingegangen. Der Unterricht erfolgt dabei in Verzahnung mit Inhalten der Mathematik und Physik. In der Sekundarstufe II bietet das Lise-Meitner-Gymnasium für die eigenen Schülerinnen und Schüler in allen Jahrgangsstufen jeweils einen Grundkurs in Informatik an. Hinzu kommen gemeinsame Grund- und Leistungskurse mit dem Friedrich-Spee-Gymnasium entsprechend dem Wahlverhalten der Schüler. Um insbesondere Schülerinnen und Schülern gerecht zu werden, die in der Sekundarstufe I keinen Informatikunterricht besucht haben, wird in Kursen der Einführungsphase Wert darauf gelegt, dass keine Vorkenntnisse aus der Sekundarstufe I zum erfolgreichen Durchlaufen des Kurses erforderlich sind.

Zurzeit besteht die Fachschaft Informatik des Lise-Meitner-Gymnasiums aus zwei Lehrkräften, denen zwei Computerräume mit 16 bzw. 19 Computerarbeitsplätzen zur Verfügung stehen. Alle Arbeitsplätze sind an das schulinterne Rechnernetz angeschlossen, so dass die Schülerinnen und Schüler Zugriff auf ihre eigene Daten und die Recherchemöglichkeiten im Internet haben. Außerdem stehen den Schülerinnen und Schülern im Selbstlernzentrum sechs PC-Arbeitsplätze zur Verfügung. Der Unterricht erfolgt ausschließlich in Doppelstunden.

2 Entscheidungen zum Unterricht

2.1 Eingesetzte Software und Schulbücher

Der Unterricht der Sekundarstufe II wird mit Hilfe der Programmiersprache Java durchgeführt. Zur Erstellung der Javaprogramme werden die didaktische Programmierumgebung BlueJ und der ebenfalls für den Bildungsbereich erstellte Javaeditor genutzt. Beide Programme sind im Vergleich zu den professionellen Programmierumgebungen Eclipse und Netbeans deutlich einfacher aufgebaut. Außerdem kommt die BlueJ-Erweiterung Greenfoot zum Einsatz, die eine übersichtliche Klassenbibliothek bietet, um grafisch ansprechende Animationen und Spiele entwickeln zu können. Wesentliche Vorteile von BlueJ sind das interaktive Erzeugen von Objekten und die Möglichkeit Zustände einzelner Objekte während der Laufzeit zu untersuchen. Dadurch wird das Erstellen von Testprogrammen stark vereinfacht. Ein Nachteil von BlueJ ist das Fehlen geeigneter Werkzeuge zur Erstellung von grafischen Oberflächen. Deshalb wird zusätzlich mit dem Javaeditor gearbeitet.

In der Einführungsphase ist zur Zeit kein Lehrbuch eingeführt, in der Qualifikationsphase wird mit dem Buch Informatik 2 bzw. Informatik 3 aus dem. Schöninghverlag gearbeitet. Die Einführung des überarbeiteten Band 1 aus der Reihe ist für das Schuljahr 2015/16 vorgesehen.

2.2 Unterrichtsvorhaben

Im Rahmen der nachfolgend beschriebenen Unterrichtsvorhaben werden alle im Kernlehrplan angeführten Kompetenzen abgedeckt. Dokumentiert wird die Umsetzung auf zwei Ebenen, der Übersichts- und der Konkretisierungsebene.

Im „Übersichtsraster Unterrichtsvorhaben“ (Abschnitt 2.2.1) wird die für alle Lehrerinnen und Lehrer gemäß Fachkonferenzbeschluss verbindliche Verteilung der Unterrichtsvorhaben dargestellt. Das Übersichtsraster dient dazu, den Kolleginnen und Kollegen einen schnellen Überblick über die Zuordnung der Unterrichtsvorhaben zu den einzelnen Jahrgangsstufen sowie den im Kernlehrplan genannten Kompetenzen, Inhaltsfeldern und inhaltlichen Schwerpunkten zu verschaffen. Der ausgewiesene Zeitbedarf versteht sich als grobe Orientierungsgröße, die nach Bedarf über- oder unterschritten werden kann. Um Freiraum für Vertiefungen, besondere Schülerinteressen, aktuelle Themen bzw. die Erfordernisse anderer besonderer Ereignisse (z.B. Praktika, Kursfahrten o.ä.) zu erhalten, wurden im Rahmen dieses schulinternen Lehrplans etwa $\frac{3}{4}$ der Bruttounterrichtszeit verplant.

Während der Fachkonferenzbeschluss zum „Übersichtsraster Unterrichtsvorhaben“ zur Gewährleistung vergleichbarer Standards sowie zur Absicherung von Lerngruppenübertritten und Lehrkraftwechseln für alle Mitglieder der Fachkonferenz bindend ist, haben die Ausweisung „konkretisierter Unterrichtsvorhaben“ (Kapitel 2.1.2) Beispiele und Materialien, einen empfehlenden Charakter.

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

2.2.1 Übersichtsraster Unterrichtsvorhaben

I) Einführungsphase

Einführungsphase	
<p><u>Unterrichtsvorhaben E-I</u></p> <p>Thema: <i>Was ist Informatik? – Überblick über die einzelnen Teilgebiete der Informatik</i></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: 4 Stunden</p>	<p><u>Unterrichtsvorhaben E-II</u></p> <p>Thema: <i>Grundlagen der objektorientierten Analyse, Modellierung und Implementierung</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Modellieren - Implementieren - Darstellen und Interpretieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Daten und ihre Strukturierung - Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Objekte und Klassen - Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: 10 Stunden</p>
<p><u>Unterrichtsvorhaben E-III</u></p> <p>Thema: <i>Algorithmische Grundstrukturen in Java</i></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 - 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: 20 Stunden</p>	<p><u>Unterrichtsvorhaben E-IV</u></p> <p>Thema: <i>Javaprogramme mit grafischen Benutzeroberflächen</i></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 - 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: 10 Stunden</p>

Einführungsphase	
<p><u>Unterrichtsvorhaben E-V</u></p> <p>Thema: <i>Digitale Daten – Einführung in die Grundlagen, Anwendungsgebiete und Verarbeitung binärer Codierung</i></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"> - Binäre Codierung und Verarbeitung - Besondere Eigenschaften der digitalen Speicherung und Verarbeitung von Daten <p>Zeitbedarf: 10 Stunden</p>	<p><u>Unterrichtsvorhaben E-VI</u></p> <p>Thema: <i>Modellierung und Implementierung von Klassen- und Objektbeziehungen eines Spiels</i></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 <p>Zeitbedarf: 8 Stunden</p>
<p><u>Unterrichtsvorhaben E-VII</u></p> <p>Thema: <i>Such- und Sortieralgorithmen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Modellieren - Darstellen und Interpretieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - 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 <p>Zeitbedarf: 8 Stunden</p>	<p><u>Unterrichtsvorhaben E-VIII</u></p> <p>Thema: <i>Leben in der digitalen Welt – Immer mehr Möglichkeiten und immer mehr Gefahren!?</i></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: 10 Stunden</p>
Summe Einführungsphase 80 U.-Std.	

II) Qualifikationsphase (Q1 und Q2) - GRUNDKURS

Qualifikationsphase 1	
<p><u>Unterrichtsvorhaben Q1-I</u></p> <p>Thema: <i>Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Modellieren - Implementieren - Darstellen und Interpretieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten - Informatiksysteme <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Objekte und Klassen - Analyse, Entwurf und Implementierung von Algorithmen - Syntax und Semantik einer Programmiersprache - Nutzung von Informatiksystemen <p>Zeitbedarf: 12 Stunden</p>	<p><u>Unterrichtsvorhaben Q1-II</u></p> <p>Thema: <i>Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Modellieren - Implementieren - Darstellen und Interpretieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Daten und ihre Strukturierung - Algorithmen - 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: 22 Stunden</p>

Qualifikationsphase 1	
<p><u>Unterrichtsvorhaben Q1-III</u></p> <p>Thema: <i>Suchen und Sortieren auf linearen Datenstrukturen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Modellieren - Implementieren - Darstellen und Interpretieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Algorithmen - 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: 16 Stunden</p>	<p><u>Unterrichtsvorhaben Q-IV</u></p> <p>Thema: <i>Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Modellieren - Implementieren - Darstellen und Interpretieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten - Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Datenbanken - Algorithmen in ausgewählten informatischen Kontexten - Syntax und Semantik einer Programmiersprache - Sicherheit <p>Zeitbedarf: 20 Stunden</p>
<p><u>Unterrichtsvorhaben Q1E-V</u></p> <p>Thema: <i>Sicherheit und Datenschutz in Netzstrukturen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Darstellen und Interpretieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Informatiksysteme - Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Einzelrechner und Rechnernetzwerke - Sicherheit - Nutzung von Informatiksystemen, Wirkungen der Automatisierung <p>Zeitbedarf: 10 Stunden</p>	
Summe Qualifikationsphase 1: 80 U.-Std.	

Qualifikationsphase 2	
<p><u>Unterrichtsvorhaben Q2-I</u></p> <p>Thema: <i>Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Modellieren - Implementieren - Darstellen und Interpretieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Objekte und Klassen - Analyse, Entwurf und Implementierung von Algorithmen - Algorithmen in ausgewählten informatischen Kontexten - Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: 24 Stunden</p>	<p><u>Unterrichtsvorhaben Q2-II</u></p> <p>Thema: <i>Endliche Automaten und formale Sprachen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Modellieren - Darstellen und Interpretieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Endliche Automaten und formale Sprachen <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Endliche Automaten - Grammatiken regulärer Sprachen - Möglichkeiten und Grenzen von Automaten und formalen Sprachen <p>Zeitbedarf: 20 Stunden</p>
<p><u>Unterrichtsvorhaben Q3-III</u></p> <p>Thema: <i>Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - 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 - Grenzen der Automatisierung <p>Zeitbedarf: 10 Stunden</p>	
Summe Qualifikationsphase 2: 54 U.-Std.	

2.1.2 Konkretisierte Unterrichtsvorhaben

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

Die Schülerinnen und Schüler

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

I) Einführungsphase

Unterrichtsvorhaben EF-I

Thema: Was ist Informatik? – Überblick über die einzelnen Teilgebiete 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 und den Bereichen der Wissenschaft Informatik zugeordnet. Das Unterrichtsvorhaben ist so strukturiert, dass die Schülerinnen und Schüler anhand bekannter Alltagstechnik die Grundideen fundamentaler informatischer Konzepte (Inhaltsfelder) zuordnen und nachvollziehen können.

Ausgehend von dem bekannten Bedienungs- und Funktionalitätswissen eines Smartphones werden technischen Bestandteile, 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 Smartphone 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: 4 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, 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).	Als Anschauungsmaterial bieten sich Smartphones an.
2. Der kompetente Umgang mit dem Schulnetzwerk (a) Erstellen und Anlegen von Ordnerstrukturen (b) Sortieren von Dateien und Ordern (c) Eingabe von Befehlen über Eingabeaufforderung (d) Einzelrechner und Netzwerk (e) Sicherheit und Datenschutz		Benutzer- und Datenschutzbestimmungen der Schule

Unterrichtsvorhaben EF-II

Thema: Grundlagen der objektorientierten Analyse, Modellierung und Implementierung

Leitfragen: Wie lassen sich Gegenstandsbereiche informatisch modellieren und 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 diesen Kontext ein. Dazu werden 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 wichtige Modellierungswerkzeuge wie Objektdiagramme und Klassendiagramme eingeführt.

Im Anschluss wird die objektorientierte Analyse für ein BlueJ-Projekt mit grafischer Ausgabe durchgeführt („Pferdekoppel“). Die vom Projekt vorgegebenen Klassen werden von Schülerinnen und Schülern in Teilen analysiert und entsprechende Objekte werden anhand einfacher Problemstellungen erprobt. Anschließend implementieren und testen die Schülerinnen und Schüler einfache Programme. Wichtig ist, dass den Schülerinnen und Schülern von Anfang an bewusst wird, dass ein Computerprogramm aus objektorientierter Sicht aus Klassen besteht, nach deren Angaben während der Laufzeit Objekten erzeugt werden, die miteinander interagieren. Zuerst liegt der Focus auf dem Klassen-Konzept (Unterscheidung zwischen Klasse und Objekt, Attribute, Methoden, Objektidentität und Objektzustand). Anschließend folgt mit der Vererbung das zweite wichtige Konzept der OOP (Ober- und Unterklasse, Generalisierung und Spezialisierung). Außerdem

werden die primitiven Datentypen int, float (double) und boolean sowie Referenzdatentypen eingeführt.

Zu diesem Zeitpunkt wird auf die Verwendung von Kontrollstrukturen verzichtet, so dass der Quellcode aus einer rein linearen Sequenz besteht. um eine Fokussierung auf die Grundlagen der Objektorientierung zu ermöglichen, ohne dass algorithmische Probleme ablenken.

Zeitbedarf: 8 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Identifikation von Objekten und Klassen (a) An dem Beispiel unterschiedlicher Pferde werden Objekte und Klassen im Sinne der objektorientierten Modellierung eingeführt. (b) Objekte werden durch Objektdiagramme, Klassen durch Klassendiagramme dargestellt. (c) 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 	Buch, Kapitel 2 Einführung in die Objektorientierung 2.1 Objektorientierte Modellierung
2. Analyse von Objekten und Klassen im BlueJ-Projekt (a) Schritte der objektorientierten Analyse, Modellierung und Implementation (b) Analyse und Erprobung der Objekte im BlueJ-Projekt		BlueJ-Projekt „Pfedekoppel“ <ul style="list-style-type: none"> – Von der Realität zu Objekten – Von den Objekten zu Klassen – Objekte inspizieren, Methoden aufrufen Objektidentität und Objektzustand
3. Implementierung einfacher Aktionen in BlueJ (a) Quelltext einer Java-Klasse (b) Implementation eigener Methoden (c) Programme übersetzen (Aufgabe des Compilers) und testen		Erweiterung des Projekts in BlueJ <ul style="list-style-type: none"> – Methoden schreiben – Programme übersetzen und testen
4. Implementierung eigener Klassen unter Verwendung der Vererbung		Unterklasse <i>Springpferd</i>

Unterrichtsvorhaben EF-III

Thema: Algorithmische Grundstrukturen in Java

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

Vorhabenbezogene Konkretisierung:

Ziel dieses Unterrichtsvorhabens ist es, das Verhalten von Objekten flexibel zu programmieren. Ein erster Schwerpunkt liegt dabei auf der Erarbeitung von Kontrollstrukturen. Dabei lernen die Schülerinnen und Schüler zwei unterschiedlichen Konzepte kennen: Wiederholungen und bedingte Anweisungen. Um eine gedankliche und begriffliche Trennung zu fördern, werden diese Kontrollstrukturen in unterschiedlichen Projekten eingeführt. Während Schleifen noch zur Weiterentwicklung des zuvor erstellte Pferdeprogramms benötigt werden, kommen bedingte Anweisungen im folgenden Projekt mit den Baukastenklassen zum Einsatz.

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. Zum Einüben des Umgangs mit Variablen wird in einem weiteren Projekt eine die Klasse Bruch modelliert und implementiert, die Operationen zur Bruchrechnung zur Verfügung stellt.

Zeitbedarf: 18 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Algorithmen (a) Wiederholungen (While-Schleife)	Die Schülerinnen und Schüler <ul style="list-style-type: none"> – analysieren und erläutern einfache Algorithmen und Programme (A), – entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M), – 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). 	Arbeitsblätter: – Schleifen BlueJ-Projekt „Pferderennen“
(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		Arbeitsblätter: – Bedingte Anweisungen – Logische Funktionen BlueJ-Projekt „Siedlung“
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		Arbeitsblätter: – lokale Variablen – Methoden und Parameter BlueJ-Projekt „Bruchrechnung“

Unterrichtsvorhaben EF-IV

Thema: *Javaprogramme mit grafische Benutzeroberflächen*

Leitfragen: Wie sind Javaprogramme mit grafischer Oberfläche aufgebaut? Wie reagieren Programme auf Benutzereingaben?

Vorhabenbezogene Konkretisierung:

Das Unterrichtsvorhaben hat die Erstellung von grafischen Benutzerschnittstellen für Javaprogrammen mit Hilfe einer visuellen Entwicklungsumgebung zum Inhalt. Wichtig ist diese Sequenz einerseits, da heute fast alle Anwendungsprogramme mit grafische Schnittstellen ausgestattet sind.

Andererseits lernen die Schülerinnen und Schüler mit dem Objektinspektor ein wichtiges Werkzeug zur Anpassung von grafischen Komponenten während des Programmentwurfs kennen.

Der vom Javaeditor generierte Code kann zu diesem Zeitpunkt zwar noch nicht vollständig erklärt werden, jedoch bietet es sich an einigen Stellen auf Details im Quelltext einzugehen. So sollte auf die main-Methode hingewiesen werden, die den Einstiegspunkt eines jeden Javaprogramms darstellt. Auch die Erstellung von jar-Archiven kann thematisiert werden. Zum Abschluss soll ein Bruchrechner unter Zuhilfenahme der bereits im Unterrichtsvorhaben III implementierten Bruchklasse erstellt werden.

Zeitbedarf: 8 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Programme mit GUI (a) Swing-Komponenten (b) Zugriff auf Komponenten mit dem Objektinspektor (c) Getter- und Settermethoden (d) Umwandlung von Datentypen (String und int)	Die Schülerinnen und Schüler <ul style="list-style-type: none"> – analysieren und erläutern einfache Algorithmen und Programme (A), – ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen zu (M), – implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I), 	ABs zum Javaeditor mit Übungsaufgaben
2. Prinzipien objektorientierter Programmierung <ul style="list-style-type: none"> • Geheimnisprinzip • Wiederverwendbarkeit 	<ul style="list-style-type: none"> – 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). 	Erstellung einer Oberfläche für einen Bruchrechner (mit der Klasse <i>Bruch</i>)

Unterrichtsvorhaben EF-V

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: 8 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 <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) 	<i>Exkurs „Analog und Digital“</i>
2. Der Umgang mit binärer Codierung von Informationen (a) Das binäre (und hexadezimale) Zahlensystem (b) Binäre Informationsspeicherung (c) Binäre Verschlüsselung (d) Implementation eines Binärumrechners	<ul style="list-style-type: none"> – 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) 	<i>Exkurs „Binäre Welt“</i>
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		<i>Exkurs „Arbeitsweise eines Computers“</i>

Unterrichtsvorhaben EF-VI

Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand eines Spiels

Leitfragen: Wie werden realistische Systeme anforderungsspezifisch reduziert, als Entwurf modelliert und implementiert? Wie werden Beziehungen zwischen Klassen dargestellt und implementiert?

Vorhabenbezogene Konkretisierung:

Das Unterrichtsvorhaben hat die Entwicklung von Objekt -und Klassenbeziehungen zum Schwerpunkt. Dazu wird, ausgehend von der Realität, über Objektidentifizierung und Entwurf bis hin zur Implementation ein Softwareprodukt ganzheitlich erstellt.

Zuerst identifizieren die Schülerinnen und Schüler Objekte und stellen diese dar. Aus diesen Objekten werden Klassen gebildet und ihre Beziehungen werden in einem Entwurfsdiagramm übernommen. Anschließend wird ein Exkurs unternommen, um die BlueJ-Erweiterung Greenfoot vorzustellen. Durch die Bearbeitung kleinerer Übungen machen sich die Schüler unter Verwendung der Klassendokumentation mit der Entwicklungsumgebung vertraut. Danach wird ein vorgegebenes Implementationsdiagramm des Spiels mit dem zuvor gemeinsam erstellen Entwurfsdiagramm verglichen. Zum Schluss programmieren die Schüler das Spiel weitgehend selbstständig in Javaklassen.

Zeitbedarf: 8 Stunden

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), – modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), 	<p>Demonstration des „Höher-Niedriger-Spiels“-Spiels</p>
<p>2. Exkurs: Greenfoot und die Arbeit mit der Dokumentation</p> <p>(a) Vorstellen der Klassen <i>World</i> und <i>Actor</i></p> <p>(b) Erweiterung um eigene Unterklassen</p> <p>(c) Wiederholung von Arrays</p>	<ul style="list-style-type: none"> – ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M), – ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M), – modellieren Klassen unter Verwendung von Vererbung (M), 	<p>Szenario „Sealife“</p> <p>AB „Szenarien mit Greenfoot entwerfen“</p> <p>AB „Impelentierung eines Rundenzählers“</p>
<p>3. Implementationsdiagramme als erster Schritt der Programmierung</p> <p>(a) Erweiterung des Entwurfsdiagramms um Konstruktoren und get- und set-Methoden</p> <p>(b) Festelegung von Datentypen in Java, sowie von Rückgaben und Parametern</p> <p>(a) Modellierung von Attributen als Felder</p> <p>(c) Deklaration, Instanziierung und Zugriffe auf ein Feld</p>	<ul style="list-style-type: none"> – implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), – 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). 	<p>Das „Höher-Niedriger-Spiel“</p> <p>AB „Implementationsdiagramm“</p> <p>Bilddateien „Spielkarten“</p>
<p>4. Implementierung</p> <p>(a) Klassen werden in Java-Quellcode umgesetzt</p> <p>(b) Einzelne Klassen und das Gesamtsystem werden anhand der Anforderungen und Dokumentationen auf ihre Korrektheit überprüft</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 8 Softwareprojekte</p> <p>8.1 Softwareentwicklung</p> <p>8.2 Oberflächen</p>

Unterrichtsvorhaben EF-VII**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, die zuerst in Pseudocode erarbeitet und anschließend in Java umgesetzt werden. Dabei lernen die Schülerinnen und Schüler zunächst Strategien des Suchens (lineare Suche, ggf. auch binäre Suche) und dann des Sortierens (Selection Sort, Insertion Sort und Bubble Sort) kennen. Der Projekteinstiege mit Spielkarten dienen dazu, die jeweiligen Strategien handlungsorientiert zu erkunden und die zugrundeliegenden Algorithmen in Pseudocode darzustellen. Dabei können bereits intuitive Effizienzbetrachtungen der Suchalgorithmen vorgenommen werden. Bei der anschließenden Implementierung sollen die Schülerinnen und Schüler auf die im letzten Unterrichtsvorhaben erstellten Klassen *Spielkarte* und *Croupier* zurückgreifen.

Zum Abschluss des Unterrichtsvorhabens wird die Effizienz der unterschiedlicher Sortierverfahren beurteilt. Dazu wird ein Zähler implementiert, der für jedes Verfahren die Anzahl der beim Sortieren vorgenommenen Vergleiche bestimmt. Von diesen Werten ausgehend wird unter Berücksichtigung des Zusammenhangs zwischen Feldgröße und Anzahl der Sortierschritte die Effizienz dieser Verfahren allgemein dargestellt.

Zeitbedarf: 8 Stunden**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Modellierung und Implementation von Datenansammlungen (a) Modellierung von Attributen als Felder (b) Deklaration, Instanziierung und Zugriffe auf ein Feld	Die Schülerinnen und Schüler <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) 	Sortieren und Suchen auf Feldern Das Feld – Eine Sammlung von Daten
2. Explorative Erarbeitung von Suchverfahren (a) Erkundung von Strategien für das Suchen auf unsortierten Daten, auf sortierten Daten und mithilfe einer Berechnungsfunktion. (b) Vergleich der drei Verfahren durch intuitive Effizienzbetrachtungen.	<ul style="list-style-type: none"> – ordnen Attributen lineare Datenansammlungen zu (M) 	Sortieren und Suchen auf Feldern Suchen mit System: <ul style="list-style-type: none"> – Lineare Suche – Binäre Suche

<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) Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des Speicherplatzbedarfs</p> <p>(e) Analyse eines weiteren Sortieralgorithmus (sofern nicht in (a) bereits geschehen)</p>		<p>Sortieren und Suchen auf Feldern</p> <p>Sortieren:</p> <ul style="list-style-type: none"> – Selection Sort – Insertion Sort – Bubble Sort
--	--	--

Unterrichtsvorhaben EF-VIII

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: 10 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Schriftzeichen, Rechenmaschine, Computer</p> <p>(a) Anhand von Schwerpunkten, wie z.B. Datenspeicherung sollen wichtige Entwicklungen</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> – bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A), – erläutern wesentliche Grundlagen der 	<p>Exkurs „Geschichte der Informatik“</p>

<p>der Informatik vorgestellt werden.</p> <p>(b) Anhand der unterschiedlichen Schwerpunkte sollen universelle Tendenzen der Entwicklung der Informationsverarbeitung erarbeitet werden.</p>	<p>Geschichte der digitalen Datenverarbeitung (A)</p> <p>– nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K)</p>	
<p>2. Die Informationsverarbeitung und ihre Möglichkeiten und Gefahren</p> <p>(a) Ausgehend von 1. werden Tendenzen der Entwicklung der Informatik erarbeitet</p> <p>(b) Informatik wird als Hilfswissenschaft klassifiziert, die weit über ihren originären Bereich hinaus Effizienz- und Leistungssteigerungen erzeugt</p> <p>(c) Betrachtung von Fallbeispielen (technische und organisatorische Vorteile, sowie deren datenschutzrechtlichen Nachteile)</p>		<p><i>Exkurs „Informatik und Gesellschaft“</i></p>

II) Qualifikationsphase

Unterrichtsvorhaben Q1-I:

Thema: Wiederholung der objektorientierten Modellierung und Programmierung

Leitfragen: Wie modelliert und implementiert man zu einer Problemstellung in einem geeigneten Anwendungskontext Java-Klassen inklusive ihrer Attribute, Methoden und Beziehungen? Wie kann man die Modellierung und die Funktionsweise der Anwendung grafisch darstellen?

Vorhabenbezogenen Konkretisierung:

Zu einer Problemstellung in einem Anwendungskontext soll eine Java-Anwendung entwickelt werden. Die Problemstellung soll so gewählt sein, dass für diese Anwendung die Verwendung einer abstrakten Oberklasse als Generalisierung verschiedener Unterklassen sinnvoll erscheint und eine Klasse durch eine Unterklasse spezialisiert werden kann. Um die Aufgabe einzugrenzen, können (nach der ersten Problemanalyse) einige Teile (Modellierungen oder Teile von Java-Klassen) vorgegeben werden.

Die Schülerinnen und Schülern erläutern und modifizieren den ersten Entwurf und modellieren sowie implementieren weitere Klassen und Methoden für eine entsprechende Anwendung. Klassen und ihre Beziehungen werden in einem Implementationsdiagramm dargestellt. Dabei werden

Sichtbarkeitsbereiche zugeordnet. Exemplarisch wird eine Klasse dokumentiert. Der Nachrichtenaustausch zwischen verschiedenen Objekten wird verdeutlicht, indem die Kommunikation zwischen zwei ausgewählten Objekten grafisch dargestellt wird. In diesem Zusammenhang wird das Nachrichtenkonzept der objektorientierten Programmierung wiederholt.

Zeitbedarf: 8 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
Wiederholung und Erweiterung der objektorientierten Modellierung und Programmierung durch Analyse und Erweiterung eines kontextbezogenen Beispiels (a) Analyse der Problemstellung (b) Analyse der Modellierung (Implementationsdiagramm) (c) Erweiterung der Modellierung im Implementationsdiagramm (Vererbung, abstrakte Klasse) (d) Kommunikation zwischen mindestens zwei Objekten (grafische Darstellung) (e) Dokumentation von Klassen (f) Implementierung der Anwendung oder von Teilen der Anwendung	Die Schülerinnen und Schüler <ul style="list-style-type: none"> – analysieren und erläutern objektorientierte Modellierungen (A), – beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), – modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), – ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), – modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), – implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), – nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen – wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I), – interpretieren Fehlermeldungen und korrigieren den Quellcode (I), – stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), – dokumentieren Klassen (D), – stellen die Kommunikation zwischen Objekten grafisch dar (D). 	Modellierung eines Rollenspiels

Unterrichtsvorhaben Q1-II:

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

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

Vorhabenbezogene Konkretisierung:

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

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse Queue</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse Queue</p> <p>(g) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Queue</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> – erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), – analysieren und erläutern Algorithmen und Programme (A), – beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), – ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), – ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), – modifizieren Algorithmen und Programme (I), – implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), – nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), – interpretieren Fehlermeldungen und korrigieren den Quellcode (I), – testen Programme systematisch anhand von Beispielen (I), – stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D). 	<p><i>Beispiel: Warteschlange am Bus</i></p> <p><i>Übung „Stausimulation“</i></p>

<p>2. Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse Stack</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse Stack</p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Stack</p>		<p><i>Beispiel: Bücherstapel</i></p>
<p>3. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse List</p> <p>(a) Erarbeitung der Vorteile der Klasse List im Gegensatz zu den bereits bekannten linearen Strukturen</p> <p>(b) Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse List.</p>		

Unterrichtsvorhaben Q1-III:

Thema: Suchen und Sortieren auf linearen Datenstrukturen

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

Vorhabenbezogene Konkretisierung:

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

Anschließend werden Sortierverfahren entwickelt und implementiert (ebenfalls für lineare Listen

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

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

Zeitbedarf: 16 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Suchen von Daten in Listen und Arrays (a) Lineare Suche in Listen und in Arrays (b) Binäre Suche in Arrays als Beispiel für rekursives Problemlösen (c) Untersuchung der beiden Suchverfahren hinsichtlich ihrer Effizienz (Laufzeitverhalten, Speicherbedarf)	Die Schülerinnen und Schüler <ul style="list-style-type: none"> – analysieren und erläutern Algorithmen und Programme (A), – beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), – beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A), – entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M), 	Beispiel: Karteiverwaltung
2. Sortieren in Listen und Arrays - Entwicklung und Implementierung von iterativen und rekursiven Sortierverfahren (a) Entwicklung und Implementierung eines einfachen Sortierverfahrens für eine Liste (b) Implementierung eines einfachen Sortierverfahrens für ein Feld (c) Entwicklung eines rekursiven Sortierverfahrens für ein Feld (z.B. Sortieren durch Mischen)	<ul style="list-style-type: none"> – modifizieren Algorithmen und Programme (I), – implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), – implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I), – nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), – interpretieren Fehlermeldungen und korrigieren den Quellcode (I), – testen Programme systematisch anhand von Beispielen (I), – stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D). 	
3. Untersuchung der Effizienz der Sortierverfahren „Sortieren durch direktes Einfügen“ und „Quicksort“ auf linearen Listen (a) Grafische Veranschaulichung der Sortierverfahren (b) Untersuchung der Anzahl der Vergleichsoperationen und des Speicherbedarfs bei beiden Sortierverfahren Beurteilung der Effizienz der beiden Sortierverfahren	<ul style="list-style-type: none"> – 	

Unterrichtsvorhaben Q1-IV:

Thema: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

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

Vorhabenbezogene Konkretisierung:

Ausgehend von einer vorhandenen Datenbank entwickeln Schülerinnen und Schüler für sie relevante Fragestellungen, die mit dem vorhandenen Datenbestand beantwortet werden sollen. Zur Beantwortung dieser Fragestellungen wird die vorgegebene Datenbank von den Schülerinnen und Schülern analysiert und die notwendigen Grundbegriffe für Datenbanksysteme sowie die erforderlichen SQL-Abfragen werden erarbeitet.

In anderen Anwendungskontexten müssen Datenbanken erst noch entwickelt werden, um Daten zu speichern und Informationen für die Beantwortung von möglicherweise auftretenden Fragen zur Verfügung zu stellen. Dafür ermitteln Schülerinnen und Schüler in den Anwendungssituationen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten und stellen diese in Entity-Relationship-Modellen dar. Entity-Relationship-Modelle werden interpretiert und erläutert, modifiziert und in Datenbankschemata überführt. Mit Hilfe von SQL-Anweisungen können anschließend im Kontext relevante Informationen aus der Datenbank extrahiert werden.

Ein Entity-Relationship-Diagramm kann auch verwendet werden, um die Entitäten inklusive ihrer Attribute und Relationen in einem vorgegebenen Datenbankschema darzustellen.

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

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Nutzung von relationalen Datenbanken</p> <p>(a) Aufbau von Datenbanken und Grundbegriffe</p> <ul style="list-style-type: none"> – Entwicklung von Fragestellungen zur vorhandenen Datenbank – Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, Datenbankschema <p>(b) SQL-Abfragen</p> <ul style="list-style-type: none"> – Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL (SELECT (DISTINCT) ...FROM, WHERE, AND, OR, NOT) auf einer Tabelle – Analyse und Erarbeitung von SQL-Abfragen auf einer und mehrerer Tabelle zur Beantwortung der Fragestellungen (JOIN, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, <>, >, <, >=, <=, LIKE, BETWEEN, IN, IS NULL) <p>(c) Vertiefung an einem weiteren Datenbankbeispiel</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> – erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A), – analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A), – analysieren und erläutern eine Datenbankmodellierung (A), – erläutern die Eigenschaften normalisierter Datenbankschemata (A), – bestimmen Primär- und Sekundärschlüssel (M), – ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M), – modifizieren eine Datenbankmodellierung (M), – modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M), – bestimmen Primär- und Sekundärschlüssel (M), – überführen Datenbankschemata in vorgegebene Normalformen (M), – 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><i>Beispiel Surfschule</i></p>
<p>2. Modellierung von relationalen Datenbanken</p> <p>(a) Entity-Relationship-Diagramm</p> <ul style="list-style-type: none"> – Ermittlung von Entitäten, zugehörigen Attributen, Relationen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship Diagramms – Erläuterung und Modifizierung einer Datenbankmodellierung 		

<p>(b) Entwicklung einer Datenbank aus einem Datenbankentwurf</p> <ul style="list-style-type: none">– Modellierung eines relationalen Datenbankschematas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln <p>(c) Redundanz, Konsistenz und Normalformen</p> <ul style="list-style-type: none">– Untersuchung einer Datenbank hinsichtlich Konsistenz und Redundanz in einer Anwendungssituation– Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten)		
---	--	--

Unterrichtsvorhaben Q1-V:**Thema:** Sicherheit und Datenschutz in Netzstrukturen**Leitfragen:** *Wie werden Daten in Netzwerken übermittelt? Was sollte man in Bezug auf die Sicherheit beachten?***Vorhabenbezogene Konkretisierung:**

Anschließend an das vorhergehende Unterrichtsvorhaben zum Thema Datenbanken werden der Datenbankzugriff aus dem Netz, Topologien von Netzwerken, eine Client-Server-Struktur, das TCP/IP-Schichtenmodell sowie Sicherheitsaspekte beim Zugriff auf Datenbanken und verschiedene symmetrische und asymmetrische kryptografische Verfahren analysiert und erläutert. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht runden das Unterrichtsvorhaben ab.

Zeitbedarf: 10 Stunden**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Daten in Netzwerken und Sicherheitsaspekte in Netzen sowie beim Zugriff auf Datenbanken (a) Beschreibung eines Datenbankzugriffs im Netz anhand eines Anwendungskontextes und einer Client-Server-Struktur zur Klärung der Funktionsweise eines Datenbankzugriffs (b) Netztopologien als Grundlage von Client-Server-Strukturen und TCP/IP-Schichtenmodell als Beispiel für eine Paketübermittlung in einem Netz (c) Vertraulichkeit, Integrität, Authentizität in Netzwerken sowie symmetrische und asymmetrische kryptografische Verfahren (Cäsar-, Vigenère-, RSA-Verfahren) als Methoden Daten im Netz verschlüsselt zu übertragen	Die Schülerinnen und Schüler <ul style="list-style-type: none"> – beschreiben und erläutern Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A), – analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A), – untersuchen und bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen, die Sicherheit von Informatiksystemen sowie die Einhaltung der Datenschutzbestimmungen und des Urheberrechts (A), – untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A), – nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D). 	Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben, Verschlüsselung Q1.5 - Zugriff auf Daten in Netzwerken
2. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht		Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1 5 - Datenschutz beim Videocenter, Materialblatt-Datenschutzgesetz

Unterrichtsvorhaben Q2-I:

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

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

Vorhabenbezogene Konkretisierung:

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

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

Eine Tiefensuche wird verwendet, um einen in der Baumstruktur gespeicherten Inhalt zu suchen. Zu einer Problemstellung in einem entsprechenden Anwendungskontext werden die Operationen der Datenstruktur Suchbaum thematisiert und unter der Verwendung der Klasse BinarySearchTree (der Materialien für das Zentralabitur in NRW) weitere Klassen oder Methoden in diesem Anwendungskontext modelliert und implementiert. Auch in diesem Kontext werden grafische Darstellungen der Bäume verwendet.

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

Zeitbedarf: 24 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Analyse von Baumstrukturen in verschiedenen Kontexten</p> <p>(a) Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit)</p> <p>(b) Aufbau und Darstellung von binären Bäumen anhand von Baumstrukturen in verschiedenen Kontexten</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> – erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), – analysieren und erläutern Algorithmen und Programme (A), – beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), – ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), 	<p><i>Beispiel: Ahnenbaum</i></p> <p><i>Weitere Beispiele für Anwendungskontexte für binäre Bäume:</i></p> <p><i>Beispiel: Suchbäume und Codierungsbäume</i></p>
<p>2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse <i>BinaryTree</i></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms</p> <p>(c) Erarbeitung der Klasse <i>BinaryTree</i> und beispielhafte Anwendung der Operationen</p> <p>(d) Implementierung der Anwendung oder von Teilen der Anwendung</p> <p>(e) Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf</p>	<ul style="list-style-type: none"> – ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), – modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), – verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M), – entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M), – implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), – modifizieren Algorithmen und Programme (I), – nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), 	
<p>3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse <i>BinarySearchTree</i></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramm,</p> <p>(c) grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften</p> <p>(d) Erarbeitung der Klasse <i>BinarySearchTree</i> und Einführung des Interface <i>Item</i> zur Realisierung einer geeigneten Ordnungsrelation</p> <p>(e) Implementierung der</p>	<ul style="list-style-type: none"> – interpretieren Fehlermeldungen und korrigieren den Quellcode (I), – testen Programme systematisch anhand von Beispielen (I), – stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D), – stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D). 	

(f) Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums	—	
4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen		

Unterrichtsvorhaben Q2-II:

Thema: Endliche Automaten und formale Sprachen

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

Vorhabenbezogene Konkretisierung:

Anhand kontextbezogener Beispiele werden endliche Automaten entwickelt, untersucht und modifiziert. Dabei werden verschiedene Darstellungsformen für endliche Automaten ineinander überführt und die akzeptierten Sprachen endlicher Automaten ermittelt. An einem Beispiel wird ein nichtdeterministischer Akzeptor eingeführt als Alternative gegenüber einem entsprechenden deterministischen Akzeptor.

Anhand kontextbezogener Beispiele werden Grammatiken regulärer Sprachen entwickelt, untersucht und modifiziert. Der Zusammenhang zwischen regulären Grammatiken und endlichen Automaten wird verdeutlicht durch die Entwicklung von allgemeinen Verfahren zur Erstellung einer regulären Grammatik für die Sprache eines gegebenen endlichen Automaten bzw. zur Entwicklung eines endlichen Automaten, der genau die Sprache einer gegebenen regulären Grammatik akzeptiert.

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

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Endliche Automaten</p> <p>(a) Vom Automaten in den Schülerinnen und Schülern bekannten Kontexten (Mealy-Automat) zur formalen Beschreibung eines endlichen Automaten (DEA)</p> <p>(b) Untersuchung, Darstellung und Entwicklung endlicher Automaten</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> – analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens auf bestimmte Eingaben (A), – analysieren und erläutern Grammatiken regulärer Sprachen (A), – zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A), – ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A), 	<p>Beispiele: Bonbon-Automat Akzeptor für Zahlen und Zeichenketten</p> <p>Erstellung von Automaten mit Auto-Edit</p>
<p>2. Untersuchung und Entwicklung von Grammatiken regulärer Sprachen</p> <p>(a) Erarbeitung der formalen Darstellung regulärer Grammatiken</p> <p>(b) Untersuchung, Modifikation und Entwicklung von Grammatiken</p> <p>(c) Entwicklung von endlichen Automaten zum Erkennen regulärer Sprachen die durch Grammatiken gegeben werden</p> <p>(d) Entwicklung regulärer Grammatiken zu endlichen Automaten</p>	<ul style="list-style-type: none"> – entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M), – entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M), – entwickeln zur akzeptierten Sprache eines Automaten die zugehörige Grammatik (M), – entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M), – modifizieren Grammatiken regulärer Sprachen (M), – entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M), – stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D), – ermitteln die Sprache, die ein endlicher Automat akzeptiert (D). – beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D). 	
<p>3. Grenzen endlicher Automaten</p>		

2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

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

Überfachliche Grundsätze:

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

Fachliche Grundsätze:

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

2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

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

2.3.1 Beurteilungsbereich Klausuren

Verbindliche Absprachen:

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

Instrumente:

Einführungsphase:	1 Klausur je Halbjahr Dauer der Klausur: 2 Unterrichtsstunden
Grundkurse Q 1:	2 Klausuren je Halbjahr Dauer der Klausuren: 2 Unterrichtsstunden
Grundkurse Q 2.1:	2 Klausuren Dauer der Klausuren: 3 Unterrichtsstunden
Grundkurse Q 2.2:	1 Klausur unter Abiturbedingungen

Anstelle einer Klausur kann gemäß dem Beschluss der Lehrerkonferenz in Q 1.2 eine Facharbeit geschrieben werden.

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

Kriterien

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

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

Von diesem kann aber im Einzelfall begründet abgewichen werden, wenn sich z.B. besonders originelle Teillösungen nicht durch Hilfspunkte gemäß den Kriterien des Erwartungshorizontes

abbilden lassen oder eine Abwertung wegen besonders schwacher Darstellung (APO-GOST §13 (2)) angemessen erscheint.

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

2.3.2 Beurteilungsbereich Sonstige Mitarbeit

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

Leistungsaspekte

Mündliche Leistungen

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

Praktische Leistungen am Computer

- Implementierung, Test und Anwendung von Informatiksystemen

Sonstige schriftliche Leistungen

- Arbeitsmappe zu einem durchgeführten Unterrichtsvorhaben
- Lernerfolgsüberprüfung durch kurze schriftliche Übungen
- Bearbeitung von schriftlichen Aufgaben im Unterricht

Kriterien

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

Die Bewertungskriterien stützen sich auf

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

Besonderes Augenmerk ist dabei auf

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

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

Bei Gruppenarbeiten auch auf

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

Bei Projektarbeit darüber hinaus auf

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

Grundsätze der Leistungsrückmeldung und Beratung

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

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

Die Leistungsrückmeldung kann

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

erfolgen.

3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen

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

Zusammenarbeit mit anderen Fächern

Im Informatikunterricht werden Kompetenzen anhand informatischer Inhalte in verschiedenen Anwendungskontexten erworben, in denen Schülerinnen und Schülern aus anderen Fächern Kenntnisse mitbringen können. Diese können insbesondere bei der Auswahl und Bearbeitung von Softwareprojekten berücksichtigt werden und in einem hinsichtlich der informatischen Problemstellung angemessenem Maß in den Unterricht Eingang finden.

Vorbereitung auf die Erstellung der Facharbeit

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

4 Qualitätssicherung und Evaluation

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

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

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