

Schulinterner Lehrplan des Gymnasium Norf
zum Kernlehrplan für die gymnasiale Oberstufe
am Gymnasium Norf



Informatik

Grund- und Leistungskurse

Entwurfsstand: 30.05.2018

Gymnasium Norf
Eichenallee 8
41469 Neuss

Inhalt

1 Die Fachgruppe Informatik am Gymnasium Norf.....	3
2 Entscheidungen zum Unterricht.....	4
2.1 Unterrichtsvorhaben.....	4
2.1.1 Übersichtsraster Unterrichtsvorhaben.....	5
2.1.2 Konkretisierte Unterrichtsvorhaben.....	14
2.2 Grundsätze der Leistungsbewertung und Leistungsrückmeldung.....	52
2.2.1 Beurteilungsbereich Klausuren.....	52
2.2.2 Beurteilungsbereich Sonstige Mitarbeit.....	53
3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen.....	55
3.1 Außerunterrichtliche Lernangebote, Wettbewerbe und externe Partner.....	55
3.2 Fächerübergreifende Bezüge.....	55
3.3 Vorbereitung auf die Erstellung der Facharbeit.....	55
3.4 Außerschulische Lernorte.....	55
3.5 Fachspezifischer und methodisch-didaktischer Bezug zum Schulprogramm.....	55
4 Didaktische Entscheidungen.....	56
4.1 Grundsätze der fachmethodischen und fachdidaktischen Arbeit.....	56
4.2 Lehr- und Lernmittel.....	57
4.3 Zuordnung Medienkompetenzen.....	57
5 Qualitätssicherung und Evaluation.....	58

1 Die Fachgruppe Informatik am Gymnasium Norf

Beim Gymnasium Norf handelt es sich um eine sechszügige Schule im südlichen Randbereich der Stadt Neuss mit zurzeit ca. 1200 Schülerinnen und Schülern und ca. 100 Lehrerinnen und Lehrern. Das Einzugsgebiet der Schule umfasst die südlichen Stadtteile von Neuss, die Schule versteht sich als Stadtteilschule. Auf Grund der exponierten Lage ist eine Kooperation mit anderen Schulen zur Zeit nicht möglich.

Das Fach Informatik – Technik – Medien (ITM) wird am Gymnasium Norf ab der Jahrgangsstufe 8 im Wahlpflichtbereich II (WP II) dreistündig unterrichtet und hierbei von etwa 35 bis 50 Schülerinnen und Schülern gewählt. In der zweijährigen Laufzeit dieser Kurse werden altersstufengerecht die Disziplinen Informatik, Technik und Medien auf vielfältige Weise thematisiert und verknüpft. Kern eines jeden Vorhabens ist der Lebensweltbezug der Schülerinnen und Schüler. Über die zwei Jahre soll ein möglichst breites Spektrum des Disziplinen abgebildet werden, um den Schülern ein lebendiges, realitätsnahes Bild zu vermitteln. Es wird Wert darauf gelegt, möglichst wenige Inhalte der Oberstufe vorwegzunehmen, damit die Wahl des Faches Informatik auch für die Schülerinnen und Schüler ohne Benachteiligung möglich ist, die das Fach nicht im Differenzierungsbereich belegen.

In der Jahrgangsstufe 5 wird, anders als in früheren Jahren, keine IT-Klasse, sondern eine freiwillige IT-AG angeboten, um allen Schülern im Rahmen des Machbaren die Möglichkeit zu einer frühen Begegnung mit der Informatik zu ermöglichen. Zukünftig soll diese in einem Mint-Profil mit informatischen Schwerpunkt einfließen. In altersgerechter Weise werden die Kinder mit den Grundlagen und Ideen des Faches vertraut gemacht. Wenn die personellen Möglichkeiten der Schule es gestatten, wird in den Jahrgangsstufen 6 und 7 für interessierte Schülerinnen und Schüler eine Arbeitsgemeinschaft „Robotik“ angeboten, die in den vergangenen Jahren auch bei Wettbewerben erfolgreich war.

In der Sekundarstufe II bietet das Gymnasium Norf für die eigenen Schülerinnen und Schüler in der Einführungsphase 2 bis 3 Grundkurse im Fach Informatik an. In der Qualifikationsphase kann in der Regel neben einem Grundkurs auch stets ein Leistungskurs eingerichtet werden. Um insbesondere Schülerinnen und Schülern gerecht zu werden, die in der Sekundarstufe I keinen Informatikunterricht besucht haben, sei nochmals betont, dass in den Kursen der Einführungsphase besonderer Wert darauf gelegt wird, keine Vorkenntnisse aus der Sekundarstufe I zum erfolgreichen Durchlaufen des Kurses einzufordern.

Der Unterricht der Sekundarstufe II wird mit Hilfe der Programmiersprache Java durchgeführt, wobei die Entwicklungsumgebungen Greenfoot und BlueJ, im Leistungskurs auch professionellere Entwicklungsumgebungen wie Netbeans oder Eclipse zum Einsatz kommen. Durch projektartiges Vorgehen, offene Aufgaben und Möglichkeiten, Problemlösungen zu verfeinern oder zu optimieren, entspricht der Informatikunterricht der Oberstufe in besonderem Maße den Erziehungszielen, Leistungsbereitschaft zu fördern, ohne zu überfordern.

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

Zurzeit besteht die Fachschaft Informatik des Gymnasium Norf aus drei Lehrkräften, denen zwei Computerräume mit jeweils 16 Computerarbeitsplätzen zur Verfügung stehen. Darüber hinaus können auch mit jeweils 16 Laptops bestückte Laptop-Wagen eingesetzt werden. Die Mediothek, die insbesondere Oberstufenschülern in Freistunden zur Verfügung steht, ist ebenfalls mit 15 Computerarbeitsplätzen ausgestattet.

Alle Arbeitsplätze und Laptops sind an das schulinterne Rechnernetz angeschlossen, so dass Schülerinnen und Schüler über einen Zugang zum zentralen Server der Schule alle Arbeitsplätze der drei Räume zum Zugriff auf ihre eigenen Daten, zur Recherche im Internet oder zur Bearbeitung schulischer Aufgaben verwenden können.

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

2 Entscheidungen zum Unterricht

2.1 Unterrichtsvorhaben

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

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

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

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

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

Inhalte und Kompetenzen, die ausschließlich im Leistungskurs behandelt werden müssen, sind grün markiert. Inhalte und Kompetenzen, die ausschließlich im Grundkurs behandelt werden müssen, sind blau markiert. Die Inhalte unterscheiden sich bezüglich ihres Umfangs kaum. Der Unterschied zwischen Grund- und Leistungskurs liegt vor allem in der Tiefe der behandelten Inhalte und dem damit verbundenen deutlich höheren Anforderungsniveau.

2.1.1 Übersichtsraster Unterrichtsvorhaben

I) Einführungsphase

Einführungsphase	
<p><u>Unterrichtsvorhaben E-I</u></p> <p>Thema: <i>Was macht Informatik? – Einführung in die Inhaltsfelder der Informatik</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">• Einsatz, Nutzung und Aufbau von Informatiksystemen• Wirkung der Automatisierung <p>Zeitbedarf: 3 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: 12 Stunden</p>

Einführungsphase

Unterrichtsvorhaben E-III

Thema:

Grundlagen der objektorientierten Analyse, Modellierung und Implementierung

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Implementieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Syntax und Semantik einer Programmiersprache
- Analyse, Entwurf und Implementierung einfacher Algorithmen

Zeitbedarf: 18 Stunden

Unterrichtsvorhaben E-IV

Thema:

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

Zentrale Kompetenzen:

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Binäre Codierung und Verarbeitung
- Besondere Eigenschaften der digitalen Speicherung und Verarbeitung von Daten

Zeitbedarf: 6 Stunden

Einführungsphase

Unterrichtsvorhaben E-V

Thema:

Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand von grafischen Spielen, Simulationen oder lebensnahen Anforderungsbeispielen

Zentrale Kompetenzen:

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

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Syntax und Semantik einer Programmiersprache
- Analyse, Entwurf und Implementierung einfacher Algorithmen

Zeitbedarf: 18 Stunden

Unterrichtsvorhaben E-VI

Thema:

Such- und Sortieralgorithmen anhand kontextbezogener Beispiele

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Algorithmen
- Daten und ihre Strukturierung

Inhaltliche Schwerpunkte:

- Algorithmen zum Suchen und Sortieren
- Analyse, Entwurf und Implementierung einfacher Algorithmen
- Objekte und Klassen

Zeitbedarf: 14 Stunden

Einführungsphase

Unterrichtsvorhaben E-VII

Thema:

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

Zentrale Kompetenzen:

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatik, Mensch und Gesellschaft
- Informatiksysteme

Inhaltliche Schwerpunkte:

- Wirkungen der Automatisierung
- Geschichte der automatischen Datenverarbeitung
- Digitalisierung

Zeitbedarf: 10 Stunden

Summe Zeitbedarf Einführungsphase: 81

II) Qualifikationsphase (Q1 und Q2)

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: 16 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: 24 Stunden</p>

Qualifikationsphase 1

Unterrichtsvorhaben Q1-III

Thema:

Suchen und Sortieren auf linearen Datenstrukturen

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Algorithmen
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

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

Zeitbedarf: 16 Stunden

Unterrichtsvorhaben Q1-IV

Thema:

Endliche Automaten und formale Sprachen

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Endliche Automaten und formale Sprachen

Inhaltliche Schwerpunkte:

- Endliche Automaten **und Kellerautomaten**
- Grammatiken regulärer **und kontextfreier** Sprachen
- Möglichkeiten und Grenzen von Automaten und formalen Sprachen
- Parser

Zeitbedarf: 24 Stunden

Qualifikationsphase 1

Unterrichtsvorhaben Q1-V

Thema:

Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen (Binäre Bäume)

Zentrale Kompetenzen:

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

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Zeitbedarf: 24 Stunden

Unterrichtsvorhaben Q1-VI

Thema:

Nutzung von relationalen Datenbanken in Anwendungskontexten

Zentrale Kompetenzen:

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

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Zeitbedarf: 10 Stunden

Summe Zeitbedarf Qualifikationsphase 1: 114 Stunden

Qualifikationsphase 2

Unterrichtsvorhaben Q2-I

Thema:

Modellierung von relationalen Datenbanken in Anwendungskontexten

Zentrale Kompetenzen:

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

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Zeitbedarf: 16 Stunden

Unterrichtsvorhaben Q2-II

Thema:

Sicherheit und Datenschutz in Netzstrukturen, Informatiksysteme

Zentrale Kompetenzen:

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

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

Zeitbedarf: 22 Stunden

Qualifikationsphase 2

Unterrichtsvorhaben Q2-III

Thema:

Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen (Graphen)

Zentrale Kompetenzen:

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

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Zeitbedarf: 22 Stunden

Unterrichtsvorhaben Q2-IV

Thema:

Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

Zentrale Kompetenzen:

- Argumentieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Einzelrechner und Rechnernetzwerke
- Grenzen der Automatisierung

Zeitbedarf: 16 Stunden

Summe Zeitbedarf Qualifikationsphase 2: 76 Stunden

2.1.2 Konkretisierte Unterrichtsvorhaben

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

Hinweis:

Verbindliche Festlegungen der Fachkonferenz:

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

Unterrichtliche Anregungen:

Die angeführten Beispiele, Medien und Materialien sind dagegen Vorschläge bzw. Hilfen für die Lehrkräfte. In diesen Bereichen sind Abweichungen von den vorgeschlagenen Vorgehensweisen möglich. Die Angabe der Kapitel bezieht sich auf das Lehrbuch Informatik des Schöningh-Verlags und dient vor allem der unterrichtenden Lehrkraft als Unterstützung. Das Buch kann aus dem Handapparat der Informatik ausgeliehen werden.

In der Einführungsphase werden die didaktischen Bibliotheken der Entwicklungsumgebungen Greenfoot und BlueJ verwendet. Diese stehen als kostenlose Downloads unter www.greenfoot.org und www.bluej.org zur Verfügung.

In der Qualifikationsphase werden die Unterrichtsvorhaben unter Berücksichtigung der Vorgaben für das Zentralabitur Informatik in NRW konkretisiert. Diese sind zu beziehen unter der Adresse

<https://www.standardsicherung.schulministerium.nrw.de/cms/zentralabitur-gost/faecher/fach.php?fach=15> (abgerufen: 27. 05. 2018)

I) Einführungsphase

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

Die Schülerinnen und Schüler

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

Unterrichtsvorhaben EF-I

Thema

Was macht Informatik? - Einführung in die Inhaltsfelder der Informatik

Leitfragen

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

Vorhabenbezogene Konkretisierung

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

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

Zeitbedarf

ca. 3 Stunden

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Informatiksysteme und ihr genereller Aufbau (a) Daten und ihre Strukturierung (b) Algorithmen (c) Formale Sprachen und Automaten (d) Informatiksysteme (e) Informatik, Mensch und Gesellschaft	Die Schülerinnen und Schüler <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

Unterrichtsvorhaben EF-II

Thema

Grundlagen der objektorientierten Analyse, Modellierung und Implementierung

Leitfragen

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

Vorhabenbezogene Konkretisierung

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

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

Im Anschluss wird die objektorientierte Analyse für ein vom Lehrer vorgestelltes Greenfoot-Szenario durchgeführt. Die von diesem Szenario vorgegebenen Klassen werden von Schülerinnen und Schülern in Teilen analysiert und entsprechende Objekte anhand einfacher Problemstellungen erprobt. Die Lernenden implementieren und testen einfache Programme. Die Greenfoot-Umgebung ermöglicht es, Beziehungen zwischen Klassen zu einem späteren Zeitpunkt (Kapitel 4) zu thematisieren. So kann der Fokus hier auf Grundlagen wie der Unterscheidung zwischen Klasse und Objekt, Attribute, Methoden, Objektidentität und Objektzustand gelegt werden.

Schließlich implementieren die Schülerinnen und Schüler selbstständig einfache Klassen (z.B. Batterie oder Pomme-stüte), deren Funktionalitäten sie in Greenfoot testen und mit dem Objektinspektor überprüfen können.

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

Zeitbedarf

ca. 12 Stunden

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Identifikation von Objekten und Klassen (a) An einem lebensweltnahen Beispiel 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 Programmier-	Kapitel 2 Einführung in die Objektorientierung 2.1 Objektorientierte Modellierung

<p>2. Analyse von Objekten und Klassen im Greenfoot-Szenario</p> <p>(a) Schritte der objektorientierten Analyse, Modellierung und Implementation</p> <p>(b) Analyse und Erprobung der Objekte im Greenfoot-Szenario</p>	<p>sprache, auch unter Nutzung dokumentierter Klassenbibliotheken (I).</p>	<p>Kapitel 2 Einführung in die Objektorientierung</p> <p>Von der Realität zu Objekten</p> <p>Von den Objekten zu Klassen, Klassendokumentation</p> <p>Objekte inspizieren</p> <p>Methoden aufrufen</p> <p>Objektidentität und Objektzustand</p>
<p>3. Implementierung einfacher Aktionen in Greenfoot</p> <p>(a) Quelltext einer Java-Klasse</p> <p>(b) Implementation eigener Methoden, Dokumentation mit JavaDoc</p> <p>(c) Programme übersetzen (Aufgabe des Compilers) und testen</p>		<p>Kapitel 2 Einführung in die Objektorientierung</p> <p>2.3 Programmierung in Greenfoot</p> <p>Methoden schreiben</p> <p>Programme übersetzen und testen</p>

Unterrichtsvorhaben EF-III

Thema

Algorithmische Grundstrukturen in Java

Leitfragen

Wie lassen sich Aktionen von Objekten flexibel realisieren?

Vorhabenbezogene Konkretisierung

Das Ziel dieses Unterrichtsvorhabens besteht darin, das Verhalten von Objekten flexibel zu programmieren. Ein erster Schwerpunkt liegt dabei auf der Erarbeitung von Kontrollstrukturen. An Beispielen aus dem Alltag (z.B. Kochen) werden Algorithmen unabhängig von einer Programmiersprache zunächst umgangssprachlich, daran anschließend formal (in Pseudocode) beschrieben. 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. Als weitere Darstellungsform für Algorithmen wird der Programmablaufplan eingeführt.

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

ca. 18 Stunden

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Algorithmen (a) Wiederholungen (While-Schleife) (b) bedingte Anweisungen (c) Verknüpfung von Bedingungen durch die logischen Funktionen UND, ODER und NICHT (d) Systematisierung des Vorgehens zur Entwicklung von Algorithmen zur Lösung komplexerer Probleme	Die Schülerinnen und Schüler <ul style="list-style-type: none">• analysieren und erläutern einfache Algorithmen und Programme (A),• entwerfen einfache Algorithmen und stellen sie umgangssprachlich, formal (Pseudocode) und grafisch (Programmablaufplan) dar (M),• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen zu (M),• modifizieren einfache Algorithmen und Programme (I),	Kapitel 3 Algorithmen 3.1 Wiederholungen 3.2 Bedingte Anweisungen 3.3 Logische Funktionen 3.4 Algorithmen entwickeln
2. Variablen und Methoden (a) Implementierung eigener Methoden mit lokalen Variablen, auch zur Realisierung einer Zählschleife (b) Implementierung eigener Methoden mit Parameterübergabe und/oder Rückgabewert (c) Implementierung von Konstruktor (d) Realisierung von Attributen	<ul style="list-style-type: none">• 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),	Kapitel 4 Variablen und Methoden 4.1 lokale Variablen 4.2 Methoden 4.3 Attribute

	<ul style="list-style-type: none">• interpretieren Fehlermeldungen und korrigieren den Quellcode (!).	
--	---	--

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 Zahlensystem mit arithmetischen und logischen Operationen an und codieren Zeichen binär. Als praktische Anwendung lernen die Schüler den Huffman-Code kennen.

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

Zeitbedarf

ca. 6 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)• 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)	Exkurs „Analog und Digital“
2. Der Umgang mit binärer Codierung von Informationen (a) Das binäre (und hexadezimale) Zahlensystem (b) Binäre Informationsspeicherung (c) Binäre Verschlüsselung (d) Implementation eines Binärrechners		Exkurs „Binäre Welt“, Huffman-Code
3. Aufbau informatischer Systeme (a) Identifikation des EVA-Prinzips als grundlegende Arbeitsweise infor-		Exkurs „Arbeitsweise eines Computers“

matischer Systemen (b) 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 von grafischen Spielen, Simulationen oder lebensnahen Anforderungsbeispielen

Leitfragen

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

Vorhabenbezogene Konkretisierung

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

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

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

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

Zeitbedarf

18 Stunden

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Umsetzung von Anforderungen in Entwurfsdiagramme (a) Aus Anforderungsbeschreibungen werden Objekte mit ihren Eigenschaften identifiziert (b) Gleichartige Objekte werden in Klassen (Entwurf) zusammengefasst und um Datentypen und Methoden erweitert	Die Schülerinnen und Schüler <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),	Kapitel 6 Klassenentwurf 6.1. Von der Realität zum Programm 6.2. Objekte identifizieren 6.3. Klassen und Beziehungen entwerfen
2. Implementationsdiagramme als erster Schritt der Programmierung (a) Erweiterung des Entwurfsdiagramms um Konstruktoren und	<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	Kapitel 6 Klassenentwurf 6.4 Klassen und Beziehungen implementieren

<p>get- und set-Methoden</p> <p>(b) Festlegung von Datentypen in Java, sowie von Rückgaben und Parametern</p> <p>(c) Entwicklung von Klassendokumentationen</p> <p>(d) Erstellung von Sequenzdiagrammen als Vorbereitung Vorbereitung für die Programmierung</p>	<p>Sichtbarkeitsbereich zu (M),</p> <ul style="list-style-type: none"> • modellieren Klassen unter Verwendung von Vererbung (M), • 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), 	<p>6.5 Vererbung</p>
<p>3. Programmierung anhand der Dokumentation und des Implementations- und Sequenzdiagrammes</p> <p>(a) Klassen werden in Java-Quellcode umgesetzt</p> <p>(b) Das Geheimnisprinzip wird umgesetzt</p> <p>(c) Einzelne Klassen und das Gesamtsystem werden anhand der Anforderungen und Dokumentationen auf ihre Korrektheit überprüft.</p>	<ul style="list-style-type: none"> • analysieren und erläutern einfache Algorithmen und Programme (A) • modifizieren einfache Algorithmen und Programme (I), • entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M). • stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D) 	<p>Kapitel 6 Klassenentwurf</p> <p>6.4 Klassen und Beziehungen implementieren</p> <p>6.5 Vererbung</p>
<p>4. Vererbungsbeziehungen</p> <p>(a) Das Grundprinzip der Vererbung wird erarbeitet</p> <p>(b) Die Vorteile der Vererbungsbeziehungen werden diskutiert</p> <p>(c) Vererbung wird implementiert</p>		<p>Kapitel 6 Klassenentwurf</p> <p>6.5 Vererbung</p>
<p>5. Softwareprojekt</p> <p>(a) Analyse und Dekonstruktion eines Spiels (Modelle, Quelltexte)</p> <p>(b) Erweiterung des Spiels um weitere Funktionalitäten</p> <p>(c) Modellierung eines Spiels aufgrund einer Anforderungsbeschreibung, inklusive einer grafischen Benutzeroberfläche</p> <p>(d) (arbeitsteilige) Implementation des Spiels</p>		<p>Kapitel 8 Softwareprojekte</p> <p>8.1 Softwareentwicklung</p> <p>8.2 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. Die Implementierung ausgewählter Sortieralgorithmen nach vorliegendem Pseudocode vertieft die Fähigkeiten der Schülerinnen und Schüler im Bereich der Programmierung von Algorithmen in Java. Die Analyse des Quellcodes eines Sortieralgorithmus und seine visuelle Demonstration schulen die Fähigkeiten der Codeanalyse.

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) 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

14 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)• beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeit und Speicherplatzbedarf (A)• ordnen Attributen lineare Datenansammlungen zu (M)• implementieren ausgewählte Sortieralgorithmen (I)	Kapitel 7 Sortieren und Suchen auf Feldern 7.1 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.		Kapitel 7 Sortieren und Suchen auf Feldern Projekteinstieg 1: Suchen 7.2 Suchen mit System <ul style="list-style-type: none">• Lineare Suche• Binäre Suche
3. Systematisierung von Algorithmen und Effizienzbetrachtungen		Kapitel 7 Sortieren und Suchen auf Feldern

<p>(a) Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode</p> <p>(b) Anwendung von Sortialgorithmen 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 Sortialgorithmus (sofern nicht in (a) bereits geschehen)</p>		<p>Projekteinstieg 2: Sortieren</p> <p>7.3 Ordnung ist das halbe Leben!? – Sortieren</p> <ul style="list-style-type: none"> • Selection Sort • Insertion Sort • Bubble Sort
---	--	--

Unterrichtsvorhaben EF-VII

Thema

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

Leitfragen

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

Vorhabenbezogene Konkretisierung

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

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

Ausgehend von dieser Betrachtung kann die aktuelle Informatik hinsichtlich ihrer Leistungsfähigkeit analysiert werden. Dabei soll herausgestellt werden, welche positiven und negativen Folgen Informatiksysteme mit sich bringen können. Dies kann mit der Auseinandersetzung eines fiktiven Szenarios aus dem Bereich des Films einhergehen.

Zeitbedarf

ca. 10 Stunden

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Schriftzeichen, Rechenmaschine, Computer (a) Anhand von Schwerpunkten, wie z.B. Datenspeicherung, Maschinen, Vernetzung sollen wichtige Entwicklungen der Informatik vorgestellt werden. (b) Anhand der unterschiedlichen Schwerpunkte sollen universelle Tendenzen der Entwicklung der Informationsverarbeitung erarbeitet werden.	Die Schülerinnen und Schüler <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)	Exkurs „Geschichte der Informatik“
2. Die Informationsverarbeitung und ihre Möglichkeiten und Gefahren (a) Ausgehend von 1. werden Tendenzen der Entwicklung der Informatik erarbeitet (b) Informatik wird als Hilfswissenschaft klassifiziert, die weit über ihren originären Bereich hinaus Effizienz- und Leistungssteigerungen erzeugt		Exkurs „Informatik und Gesellschaft“

<p>(a) Anhand von Fallbeispielen werden technische und organisatorische Vorteile, sowie deren datenschutzrechtlichen Nachteile betrachtet.</p>		
--	--	--

II) Qualifikationsphase

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

Die Schülerinnen und Schüler

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

Unterrichtsvorhaben Q1-I:

Thema

Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung

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	Beispiele, Medien, Materialien
<p>1. Wiederholung und Erweiterung der objektorientierten Modellierung und Programmierung durch Analyse und Erweiterung eines kontextbezogenen Beispiels</p> <p>(a) Analyse der Problemstellung</p> <p>(b) Analyse der Modellierung (Implementationsdiagramm)</p> <p>(c) Erweiterung der Modellierung im Implementationsdiagramm (Vererbung, abstrakte Klasse)</p> <p>(d) Kommunikation zwischen mindestens zwei Objekten (grafische Darstellung)</p> <p>(e) Dokumentation von Klassen</p> <p>(f) Implementierung der Anwendung oder von Teilen der Anwendung</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">• analysieren und erläutern objektorientierte Modellierungen (A),• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),• modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M),• ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M),• modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),• implementieren Klassen in einer Programmiersprache auch unter	<p>„Objektkommunikation in Omok“:</p> <p>Implementierung / Erweiterung des Spiels „Vier gewinnt“</p>

	<p>Nutzung dokumentierter Klassenbibliotheken (I),</p> <ul style="list-style-type: none">• nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),• wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I),• 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).	
--	--	--

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	Beispiele, Medien, Materialien
<p>1. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse Queue</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse Queue</p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Queue</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">• erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A),• analysieren und erläutern Algorithmen und Programme (A),• beurteilen die syntaktische Korrektheit und 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	<p>Beispiel: Patientenwarteschlange (jeder kennt seinen Nachfolger bzw. alternativ: seinen Vorgänger)</p> <p>Sobald ein Patient in einer Arztpraxis eintrifft, werden sein Name und seine Krankenkasse erfasst. Die Verwaltung der Patientenwarteschlange geschieht über eine Klasse, die hier als Wartezimmer bezeichnet wird. Wesentliche Operationen sind das „Hinzufügen“ eines Patienten und das „Entfernen“ eines Patienten, wenn er zur Behandlung gerufen wird.</p> <p>Die Simulationsanwendung stellt eine GUI zur Verfügung, legt ein Wartezimmer an und steuert die Abläufe. Wesentlicher Aspekt des Projektes ist die Modellierung des Wartezimmers mit Hilfe der Klasse Queue.</p> <p>Anschließend wird der Funktionsumfang der Anwendung erweitert: Patienten können sich zusätzlich in die Warteschlange zum Blutdruckmessen einreihen. Objekte werden von zwei Schlangen</p>

	<p>von dynamischen Datenstrukturen (I),</p> <ul style="list-style-type: none"> • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen (I), • stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D). 	<p>verwaltet.</p> <p>Materialien:</p> <p>Ergänzungsmaterialien zum Lehrplannavigators Unterrichtsvorhaben Q1.2 – Warteschlange</p> <p>(Download Q1-II.1)</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>Beispiel: Heftstapel</p> <p>In einem Heftstapel soll das Heft einer Schülerin gefunden werden.</p> <p>oder</p> <p>Beispiel: Kisten stapeln</p> <p>In einem Stapel nummerierter Kisten soll eine bestimmte Kiste gefunden und an einen Kunden geliefert werden. Dazu müssen Kisten auf verschiedene Stapel gestapelt und wieder zurückgestellt werden.</p>
<p>3. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse 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>		<p>Beispiel: Abfahrtslauf</p> <p>Bei einem Abfahrtslauf kommen die Skifahrer nacheinander an und werden nach ihrer Zeit in eine Rangliste eingeordnet. Diese Rangliste wird in einer Anzeige ausgegeben. Ankommende Abfahrer müssen an jeder Stelle der Struktur, nicht nur am Ende oder Anfang eingefügt werden können.</p> <p>Materialien:</p> <p>Ergänzungsmaterialien zum Lehrplannavigators Unterrichtsvorhaben Q1.2 - Listen</p> <p>(Download Q1-II.2)</p>
<p>4. Vertiefung - Anwendungen von Listen, Stapeln oder Schlangen in mindestens einem weiteren Kontext</p>		<p>Beispiel: Skispringen</p> <p>Ein Skispringen hat folgenden Ablauf: Nach dem Sprung erhält der Springer eine Punktzahl und wird nach dieser Punktzahl in eine Rangliste eingeordnet. Die besten 30 Springer qualifizieren sich für den zweiten Durchgang. Sie starten in umgekehrter Reihenfolge gegenüber der Platzierung auf der Rangliste. Nach dem Sprung erhält der Springer wiederum eine Punktzahl und wird nach der Gesamtpunktzahl aus beiden Durchgängen in die endgültige Rangliste eingeordnet.</p> <p>Beispiel: Rangierbahnhof</p> <p>Auf einem Güterbahnhof gibt es drei</p>

		<p>Gleise, die nur zu einer Seite offen sind. Wagons können also von einer Seite auf das Gleis fahren und nur rückwärts wieder hinausfahren. Die Wagons tragen Nummern, wobei die Nummer jedoch erst eingesehen werden kann, wenn der Wagon der vorderste an der offenen Gleisseite ist. (Zwischen den Wagons herumzuturnen, um die anderen Wagonnummern zu lesen, wäre zu gefährlich.) Zunächst stehen alle Wagons unsortiert auf einem Gleis. Ziel ist es, alle Wagons in ein anderes Gleis zu fahren, so dass dort die Nummern der Wagons vom Gleisende aus aufsteigend in richtiger Reihenfolge sind. Zusätzlich zu diesen beiden Gleisen gibt es ein Abstellgleis, das zum Rangieren benutzt werden kann.</p> <p>Beispiel: Autos an einer Ampel zur Zufahrtsregelung</p> <p>Es soll eine Ampel zur Zufahrtsregelung in Java simuliert werden. An einem geradlinigen, senkrecht von unten nach oben verlaufenden Straßenstück, das von Autos nur einspurig in eine Richtung befahren werden kann, ist ein Haltepunkt markiert, an dem die Ampel steht. Bei einem Klick auf eine Schaltfläche mit der Aufschrift „Heranfahren“ soll ein neues Auto an den Haltepunkt heranfahren bzw. bis an das letzte Auto, das vor dem Haltepunkt wartet. Grünphasen der Ampel werden durch einen Klick auf eine Schaltfläche mit der Aufschrift „Weiterfahren“ simuliert. In jeder Grünphase darf jeweils nur ein Auto weiterfahren. Die anderen Autos rücken nach.</p> <p>Materialien:</p> <p>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1-II.3 – Anwendungen für lineare Datenstrukturen (Download Q1-II.3)</p>
--	--	---

Unterrichtsvorhaben Q1-III:

Thema

Suchen und Sortieren auf linearen Datenstrukturen

Leitfrage

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

Vorhabenbezogene Konkretisierung

Zunächst werden übergreifende Algorithmeigenschaften wie Korrektheit, Effizienz und Verständlichkeit erarbeitet und Schritte der Algorithmenentwicklung wiederholt.

Als besondere Struktur von Algorithmen wird die Rekursion gegenüber der Iteration abgegrenzt. Rekursive Algorithmen werden analysiert und selbst entwickelt.

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

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

Abschließend werden verschiedene Sortierverfahren hinsichtlich der Anzahl der benötigten Vergleichsoperationen und des Speicherbedarfs beurteilt **und der best, worst und average case begründet einer Komplexitätsklasse zugeordnet.**

Zeitbedarf

16 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Rekursive und iterative Algorithmen	Die Schülerinnen und Schüler	Beispiel: Fakultät oder Fibonacci-Folge
2. 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)	<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 Für ein Adressverwaltungsprogramm soll eine Methode zum Suchen einer Adresse geschrieben werden. oder Beispiel: Bundesjugendspiele Die Teilnehmer an Bundesjugendspielen nehmen an drei Disziplinen teil und erreichen dort Punktzahlen. Diese werden in einer Wettkampfkarte eingetragen und an das Wettkampfbüro gegeben.

	<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). 	<p>Zur Vereinfachung sollte sich das Modell auf die drei Disziplinen „Lauf“, „Sprung“ und „Wurf“ beschränken.</p> <p>Im Wettkampfbüro wird das Ergebnis erstellt. Das Programm soll dafür zunächst den Besten einer Disziplin herausuchen können und später das gesamte Ergebnis nach gewissen Kriterien sortieren können.</p> <p>Materialien: Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.3 - Suchen und Sortieren (Download Q1-III.1)</p>
<p>3. Sortieren in Listen und Arrays - Entwicklung und Implementierung von iterativen und rekursiven Sortierverfahren</p> <p>(a) Entwicklung und Implementierung eines einfachen Sortierverfahrens für eine Liste</p> <p>(b) Implementierung eines einfachen Sortierverfahrens für ein Feld</p> <p>(c) Entwicklung eines rekursiven Sortierverfahrens für ein Feld (z.B. Sortieren durch Mischen (Mergesort))</p>		<p>Beispiel: Karteiverwaltung (s.o.) oder Beispiel: Bundesjugendspiele (s.o.) Materialien: (s.o.)</p>
<p>4. Untersuchung der Effizienz der Sortierverfahren „Insertionsort“ und „Quicksort“ auf linearen Listen</p> <p>(a) Grafische Veranschaulichung der Sortierverfahren</p> <p>(b) Untersuchung der Anzahl der Vergleichsoperationen und des Speicherbedarf bei beiden Sortierverfahren</p> <p>(c) Beurteilung der Effizienz der beiden Sortierverfahren</p> <p>(d) Zuordnung der Verfahren zu einer Komplexitätsklasse</p> <p>(e) Implementierung von Quicksort</p>		<p>Beispiel: Karteiverwaltung (s.o.) oder Beispiel: Bundesjugendspiele (s.o.) Materialien: (s.o.)</p>

Unterrichtsvorhaben Q1-IV:

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 zu 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. Kellerautomaten werden als zugehörige Akzeptoren kontextfreier Sprachen eingeführt. Die Formulierung von kontextfreien Grammatiken und die Analyse der Zugehörigkeit einer Grammatik zu einer Sprachklasse schließen das Vorhaben ab.

Die Implementierung von Parsern für reguläre und kontextfreie Sprachen wird in verschiedenen Varianten besprochen und umgesetzt.

Zeitbedarf

20 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
1. Endliche Automaten (a) Vom Automaten in den Schülerinnen und Schülern bekannten Kontexten zur formalen Beschreibung eines endlichen Automaten (b) Untersuchung, Darstellung und Entwicklung endlicher Automaten	Die Schülerinnen und Schüler <ul style="list-style-type: none">• analysieren und erläutern die Eigenschaften endlicher Automaten und Kellerautomaten einschließlich ihres Verhaltens auf bestimmte Eingaben (A),• analysieren und erläutern Grammatiken regulärer und kontextfreier Sprachen (A),• zeigen die Grenzen endlicher Automaten und Kellerautomaten sowie regulärer und kontextfreier Grammatiken im Anwendungszusammenhang auf (A),	Beispiele: Cola-Automat, Geldspielautomat, Roboter, Zustandsänderung eines Objekts „Auto“, Akzeptor für bestimmte Zahlen, Akzeptor für Teilwörter in längeren Zeichenketten, Akzeptor für Terme Materialien: Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.2 – Endliche Automaten, Formale Sprachen

		(Download Q2-II.1)
2. Untersuchung und Entwicklung regulärer Sprachen (a) Erarbeitung der formalen Darstellung regulärer Grammatiken (b) Untersuchung, Modifikation und Entwicklung von Grammatiken (c) Entwicklung von endlichen Automaten zum Erkennen regulärer Sprachen, die durch Grammatiken gegeben sind (d) Entwicklung regulärer Grammatiken zu endlichen Automaten	<ul style="list-style-type: none"> • ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A), • entwickeln und modifizieren zu einer Problemstellung endliche Automaten und Kellerautomaten (M), • entwickeln zur akzeptierten Sprache eines Automaten die zugehörige Grammatik (M), • entwickeln eine Grammatik in EBNF • entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M), • entwickeln zur Grammatik einer kontextfreien Sprache einen zugehörigen Kellerautomaten (M), 	
3. Grenzen endlicher Automaten		
4. Untersuchung und Entwicklung kontextfreier Sprachen (a) Erarbeitung der formalen Darstellung kontextfreier Grammatiken (b) Untersuchung, Modifikation und Entwicklung von Grammatiken (c) Entwicklung von Kellerautomaten zum Erkennen kontextfreier Sprachen, die durch Grammatiken gegeben sind (d) Zuordnung einer Grammatik zu einer Sprachklasse	<ul style="list-style-type: none"> • modifizieren Grammatiken regulärer und kontextfreier Sprachen (M), • entwickeln zu einer Sprache eine Grammatik, die die Sprache erzeugt (M), • überführen mittels Potenzmen-genkonstruktion einen NEA in einen DEA (M), • stellen endliche Automaten und Kellerautomaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D), 	
5. Nichtdeterministische endliche Automaten (a) Gegenüberstellung von NEA bzw. ϵ -NEA und DEA (b) Umwandlung eines NEA in einen DEA mittels Potenzmen-genkonstruktion	<ul style="list-style-type: none"> • ermitteln die Sprache, die ein endlicher Automat und ein Kellerautomat akzeptieren (D), • beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D). 	

Unterrichtsvorhaben Q1-V:

Thema

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

Leitfragen

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

Vorhabenbezogene Konkretisierung

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

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

Unter anderem sollen die verschiedenen Baumtraversierungen (Pre-, Post- und Inorder) implementiert werden. Unterschiede bezüglich der Möglichkeit, den Baum anhand der Ausgabe der Bauminhalte via Pre-, In- oder Postorder-Traversierung zu rekonstruieren, werden dabei ebenfalls angesprochen, indem die fehlende Umkehrbarkeit der Zuordnung Binärbaum \leftrightarrow 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. Um Daten in einem solchen Anwendungskontext verwalten zu können, müssen sie in eine Ordnung gebracht werden, d.h. sie müssen vergleichbar sein. Diese Vorgabe wird mithilfe des Interfaces `Item` realisiert, das alle Klassen, deren Objekte in einem Suchbaum verwaltet werden sollen, implementiert werden müssen. Auch in diesem Kontext werden grafische Darstellungen der Bäume verwendet. Klassen und Beziehungen werden in Entwurfs- und Implementationsdiagrammen dargestellt.

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

Zeitbedarf

24 Stunden

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Analyse von Baumstrukturen in verschiedenen Kontexten (a) Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit) (b) Aufbau und Darstellung von bi-	Die Schülerinnen und Schüler <ul style="list-style-type: none">• erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A),• analysieren und erläutern Algorithmen	Beispiel: Termbaum Der Aufbau von Termen wird mit Hilfe von binären Baumstrukturen verdeutlicht. oder

<p>nären Bäumen anhand von Baumstrukturen in verschiedenen Kontexten</p>	<p>men und Programme (A),</p> <ul style="list-style-type: none"> • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • modifizieren Algorithmen und Programme (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen (I), • stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D). 	<p>Beispiel: Ahnenbaum</p> <p>Die binäre Baumstruktur ergibt sich daraus, dass jede Person genau einen Vater und eine Mutter hat.</p> <p>Weitere Beispiele für Anwendungskontexte für binäre Bäume:</p> <p>Beispiel: Suchbäume (zur sortierten Speicherung von Daten)</p> <p>Alle Inhalte, die nach einer Ordnung vor dem Inhalt im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem Inhalt im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.)</p> <p>oder</p> <p>Beispiel: Entscheidungsbäume</p> <p>Um eine Entscheidung zu treffen, werden mehrere Fragen mit ja oder nein beantwortet. Die Fragen, die möglich sind, wenn die Antwort auf eine Frage mit „ja“ beantwortet wird, befinden sich im linken Teilbaum, die Fragen, die möglich sind, wenn die Antwort „nein“ lautet, stehen im rechten Teilbaum.</p> <p>oder</p> <p>Beispiel: Codierungsbäume für Codierungen, deren Alphabet aus genau zwei Zeichen besteht</p> <p>Morse hat Buchstaben als Folge von Punkten und Strichen codiert. Diese Codierungen können in einem Binärbaum dargestellt werden, so dass ein Übergang zum linken Teilbaum einem Punkt und ein Übergang zum rechten Teilbaum einem Strich entspricht. Wenn man im Gesamtbaum startet und durch Übergänge zu linken oder rechten Teilbäumen einen Pfad zum gewünschten Buchstaben sucht, erhält man die Morsecodierung des Buchstabens.</p> <p>Materialien:</p> <p>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Binärbaum</p> <p>(Download Q2-1.1)</p>
<p>2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse BinaryTree</p> <p>(a) Analyse der Problemstellung, Er-</p>		

<p>mittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms</p> <p>(c) Erarbeitung der Klasse BinaryTree und beispielhafte Anwendung der Operationen</p> <p>(d) Implementierung der Anwendung oder von Teilen der Anwendung</p> <p>(e) Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf</p>		
<p>3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse BinarySearchTree</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(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 BinarySearchTree und Einführung des Interface Item zur Realisierung einer geeigneten Ordnungsrelation</p> <p>(e) Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums</p>		<p>Beispiel: Informatikerbaum als Suchbaum</p> <p>In einem binären Suchbaum werden die Namen und die Geburtsdaten von Informatikern lexikographisch geordnet abgespeichert. Alle Namen, die nach dieser Ordnung vor dem Namen im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem Namen im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.)</p> <p>Folgende Funktionalitäten werden benötigt:</p> <ul style="list-style-type: none"> • Einfügen der Informatiker-Daten in den Baum • Suchen nach einem Informatiker über den Schlüssel Name • Ausgabe des kompletten Datenbestands in nach Namen sortierter Reihenfolge <p>Materialien:</p> <p>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Binärer Suchbaum (Download Q2-I.3)</p>
<p>4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen</p>		<p>Beispiel: Codierungsbäume (s.o.) oder Huffman-Codierung</p> <p>oder</p> <p>Beispiel: Buchindex</p> <p>Es soll eine Anwendung entwickelt werden, die anhand von Stichworten und zugehörigen Seitenzahlen ein Stichwortregister erstellt.</p> <p>Da die Stichwörter bei der Analyse des Buches häufig gesucht werden müssen,</p>

		<p>werden sie in der Klasse Buchindex als Suchbaum (Objekt der Klasse BinarySearchTree) verwaltet.</p> <p>Alle Inhalte, die nach einer Ordnung vor dem Inhalt im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem Inhalt im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.)</p> <p>oder</p> <p>Beispiel: Entscheidungsbäume (s.o.)</p> <p>oder</p> <p>Beispiel: Ternbaum (s.o.)</p> <p>oder</p> <p>Beispiel: Ahnenbaum (s.o.)</p> <p>Materialien:</p> <p>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Anwendung Binärbaum</p> <p>(Download Q2-I.4)</p>
--	--	---

Unterrichtsvorhaben Q1-VI:

Thema

Nutzung von relationalen Datenbanken in Anwendungskontexten

Leitfragen

Wie können Fragestellungen mit Hilfe einer Datenbank beantwortet werden?

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.

Dies ist der erste Teil des Themas Datenbanken, das mit dem Unterrichtsvorhaben Q2-I fortgeführt wird. Dort wird die Modellierung von Datenbanken in den Blick genommen und mit dem erworbenen Wissen dieses Unterrichtsvorhabens verknüpft.

Zeitbedarf

10 Stunden

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Nutzung von relationalen Datenbanken</p> <p>(f) 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>(g) 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>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">• analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A),• bestimmen Primär- und Sekundärschlüssel (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)	<p>Beispiel: SQLZoo</p> <p>Das Online-Tutorial von SQL-Zoo bietet die Möglichkeit, auf zahlreichen Datenbanken Abfragen auszuführen. Antworten werden ausgewertet und Lösungen visualisiert. Zudem ist es möglich, eine Abfragen zu entwickeln und deren Ergebnisse zu testen.</p> <p>Beispiel: SQLiteBrowser</p> <p>Der kostenlose SQLiteBrowser bietet die Möglichkeit, eigene Datenbanken effizient ohne SQL-Befehle anzulegen, um so eigene Abfragen zu formulieren oder Ergebnisse zu prüfen.</p>

(h) Vertiefung an einem weiteren Datenbankbeispiel		
--	--	--

Unterrichtsvorhaben Q2-I:

Thema

Modellierung von relationalen Datenbanken in Anwendungskontexten

Leitfragen

Wie entwickelt man selbst eine Datenbank für einen Anwendungskontext?

Vorhabenbezogene Konkretisierung

Dieses Unterrichtsvorhaben setzt das Thema Datenbanken, das mit Abfragen in SQL im Vorhaben Q1-VI begonnen wurde, fort.

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

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

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

Zeitbedarf

16 Stunden

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Nutzung von relationalen Datenbanken (a) Aufbau von Datenbanken und Grundbegriffe <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	Die Schülerinnen und Schüler <ul style="list-style-type: none">• erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (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),	Beispiel „Anno“: Sammeln und Ordnen von Daten für ein Computerspiel
2. Modellierung von relationalen Datenbanken (a) Entity-Relationship-Diagramm <ul style="list-style-type: none">• Ermittlung von Entitäten, zugehörigen Attributen, Relationen und Kardinalitäten in Anwendungssituatio-		Beispiel „Anno“: Modellieren einer Datenbank für ein Computerspiel

<p>nen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms</p> <ul style="list-style-type: none"> • Erläuterung und Modifizierung einer Datenbankmodellierung <p>(b) Entwicklung einer Datenbank aus einem Datenbankentwurf</p> <ul style="list-style-type: none"> • Modellierung eines relationalen Datenbankschemas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln <p>(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) 	<ul style="list-style-type: none"> • bestimmen Primär- und Sekundärschlüssel (M), • überführen Datenbankschemata in vorgegebene Normalformen (M), • stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D), • überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D). 	
--	--	--

Unterrichtsvorhaben Q2-II:

Thema

Sicherheit und Datenschutz in Netzstrukturen, Informatiksysteme

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

22 Stunden

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, 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 Netzwerk (c) Vertraulichkeit, Integrität, Authentizität in Netzwerken sowie symmetrische und asymmetrische kryptografische Verfahren (Cäsar-, Vigenère-, RSA-Verfahren) als Methoden, Daten im Netz verschlüsselt zu übertragen (d) Verifizierung einer Nachricht durch Signierung / Zertifizierung	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),	Materialien: Ergänzungsmaterialien zum Lehrplannavigators Unterrichtsvorhaben, Verschlüsselung Q1.5 - Zugriff auf Daten in Netzwerken (Download Q1-V.1)
2. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht	<ul style="list-style-type: none">• analysieren und erläutern das Prinzip des Schlüsselaustauschs nach Diffie und Hellman (A),• beschreiben das Prinzip der Signierung und Zertifizierung von Nachrichten (M)	Materialien: Ergänzungsmaterialien zum Lehrplannavigators Unterrichtsvorhaben Q1 5 - Datenschutz beim Videocenter, Materialblatt-Datenschutzgesetz

	<ul style="list-style-type: none">• implementieren eine Netzwerkanwendung mit Hilfe der Abiturklassen (M)• berechnen fehlende Parameter für das RSA-Verfahren (M),• nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zur Erschließung, Aufbereitung und Präsentation fachlicher Inhalte (D).	(Download Q1-V2)
--	---	------------------

Unterrichtsvorhaben Q2-III:

Thema

Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen (Graphen)

Leitfragen

In welchen Anwendungskontexten werden Daten mit Hilfe von Graphen verwaltet? Welche Algorithmen liefern im Anwendungskontext die gewünschten Informationen?

Vorhabenbezogene Konkretisierung

Anhand von Anwendungsbeispielen werden grundlegende Begriffe der Datenstruktur Graph eingeführt und vielfältige Einsatzmöglichkeiten vorgestellt.

Der Schwerpunkt des Vorhabens liegt auf der Entwicklung und Implementierung verschiedener Algorithmen auf Graphen unter Verwendung der für das Zentralabitur vorgegebenen Klasse Graph (Materialien für das Zentralabitur in NRW). Diese werden sowohl visualisiert und demonstriert, als auch zu großen Teilen implementiert.

Unter anderem sollen die Breiten- und Tiefensuche, der Algorithmus von Dijkstra und weitere Algorithmen (z.B. Nearest-Neighbour) thematisiert werden. Dabei sollen vor allem die Basisalgorithmen Breiten- und Tiefensuche in Varianten zur Lösung unterschiedlicher Problemstellungen modifiziert und eingesetzt werden.

Zeitbedarf

22 Stunden

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Analyse von Anwendungskontexten und Darstellung der Datenstruktur Graph</p> <p>(a) Grundlegende Begriffe (Knoten, Kante, Kantengewicht / gewichtet, gerichtet, Zyklus)</p> <p>(b) Aufbau und Darstellung von Graphen in verschiedenen Kontexten</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">• erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A),• analysieren und erläutern Algorithmen und Programme (A),• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),• ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M),• modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),	<p>Beispiel: Ariadnefaden</p> <p>Mit Hilfe der griechischen Mythologie kann am Beispiel des Ariadnefadens die Tiefensuche behandelt werden.</p> <p>Beispiel: Freundschaftsnetzwerk</p> <p>Soziale Netzwerke nutzen unter anderem Freundschaftsnetzwerke. Mit Hilfe der Breitensuche können Vorschläge für neue Freundschaften (Freunde von Freunden) unterbreitet werden.</p> <p>Beispiel Navigationssystem:</p> <p>Anhand realer Landkarten mit Entfernungsangaben kann der Algorithmus von Dijkstra erarbeitet werden.</p>

	<ul style="list-style-type: none">• verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M),• implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),• modifizieren Algorithmen und Programme (I),• nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),• testen Programme systematisch anhand von Beispielen (I),• stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D),• stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).	
--	--	--

Unterrichtsvorhaben Q2-IV:

Thema

Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

Leitfragen

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

Vorhabenbezogene Konkretisierung

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

Ausgehend von den prinzipiellen Grenzen endlicher Automaten liegt die Frage nach den Grenzen von Computern bzw. nach Grenzen der Automatisierbarkeit nahe. Mit Hilfe einer entsprechenden Java-Methode wird plausibel, dass es unmöglich ist, ein Informatiksystem zu entwickeln, dass für jedes beliebige Computerprogramm und jede beliebige Eingabe entscheidet ob das Programm mit der Eingabe terminiert oder nicht (Halteproblem). Anschließend werden Vor- und Nachteile der Grenzen der Automatisierbarkeit angesprochen und der Einsatz von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen beurteilt.

Zeitbedarf

16 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
1. Von-Neumann-Architektur und die Ausführung maschinennaher Programme (a) prinzipieller Aufbau einer von Neumann-Architektur mit CPU, Rechenwerk, Steuerwerk, Register und Hauptspeicher (b) einige maschinennahe Befehle und ihre Repräsentation in einem Binär-Code, der in einem Register gespeichert werden kann (c) Analyse und Erläuterung der Funktionsweise eines einfachen maschinennahen Programms	Die Schülerinnen und Schüler • erläutern die Ausführung eines einfachen maschinennahen Programms sowie die Datenspeicherung auf einer „Von-Neumann-Architektur“ (A), • untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A).	Beispiel: Addition von 4 zu einer eingegebenen Zahl mit einem Rechnermodell Materialien: Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 – Von-Neumann-Architektur und maschinennahe Programmierung (Download Q2-III.1)
2. Grenzen der Automatisierbarkeit (a) Vorstellung des Halteproblems (b) Unlösbarkeit des Halteproblems (c) Beurteilung des Einsatzes von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen		Beispiel: Halteproblem Materialien: Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 – Halteproblem (Download Q2-III.2)

2.2 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

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

2.2.1 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

Stufe	Anzahl der Klausuren		Dauer einer Klausur	
	GK	LK	GK	LK
EF	1 / HJ	entfällt	2 UE	entfällt
Q1	2 / HJ*	2 / HJ*	2 UE	3 UE
Q2.1	2 / HJ	2 / HJ	3 UE	4 UE
Q2.2	1, wenn Abiturfach	1	3:00h	4:15h

* 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.

2.2.2 Beurteilungsbereich Sonstige Mitarbeit

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

Verbindliche Absprachen der Fachkonferenz

- Alle Schülerinnen und Schüler führen in der Einführungsphase in Kleingruppen ein Kurzprojekt durch und fertigen dazu eine Arbeitsmappe mit Arbeitstagebuch an. Dies wird in die Note für die Sonstige Mitarbeit einbezogen.
- In der Qualifikationsphase erstellen, dokumentieren und präsentieren die Schülerinnen und Schüler in Kleingruppen ein anwendungsbezogenes Softwareprodukt. Dies wird in die Note für die Sonstige Mitarbeit einbezogen.

Leistungsaspekte

Mündliche Leistungen

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

Praktische Leistungen am Computer

- Implementierung, Test und Anwendung von Informatiksystemen

Sonstige schriftliche Leistungen

- Arbeitsmappe und Arbeitstagebuch zu einem durchgeführten Unterrichtsvorhaben
- Lernerfolgsüberprüfung durch kurze schriftliche Übungen:
- In Kursen, in denen höchstens 50% der Kursmitglieder eine Klausur schreiben, finden schriftliche Übungen mindestens einmal pro Kurshalbjahr statt, in anderen Kursen entscheidet über die Durchführung die Lehrkraft.
- Schriftliche Übungen dauern ca. 20 Minuten und umfassen den Stoff der letzten ca. 4–6 Stunden, wobei fachliches Grundlagenwissen immer vorausgesetzt werden muss.
- 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 zu legen 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.

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 erfolgen

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

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

3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen

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

3.1 Außerunterrichtliche Lernangebote, Wettbewerbe und externe Partner

Die Teilnahme am Wettbewerb „Informatik-Biber“ ist für die Stufe EF als Entscheidungshilfe für die Weiterwahl des Faches Informatik in der Q1 obligatorisch. Die Teilnahme in den anderen Jahrgangsstufen ist den Kursen nach Interesse freigestellt. Die Teilnahme am Bundeswettbewerb Informatik wird interessierten Schülerteams angeboten und im Rahmen der Möglichkeiten vom Fachlehrer begleitet.

3.2 Fächerübergreifende Bezüge

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

3.3 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.

3.4 Außerschulische Lernorte

Im Rahmen der Möglichkeiten wird in der Einführungsphase zum Unterrichtsvorhaben „Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes“ eine Exkursion zum Heinz Nixdorf Museums-Forum durchgeführt. Die außerunterrichtliche Veranstaltung wird im Unterricht vor- und nachbereitet.

3.5 Fachspezifischer und methodisch-didaktischer Bezug zum Schulprogramm

Durch seine fachimmanenten Inhalte leistet das Fach einen wichtigen Beitrag zu MINT und zur Förderung der Medienkompetenz. Diese sind fest verankert in den Unterrichtsvorhaben, wie sie oben dargestellt werden.

Die Bestrebungen, einen Informatik-Leistungskurs kontinuierlich anbieten zu können, führt zu verstärkten Bemühungen, bereits ab der 5. Klasse das Fach Informatik mit verschiedenen Angeboten in Unterricht- und AG-Form zu stärken.

4 Didaktische Entscheidungen

4.1 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

Die Lehrerkonferenz des Gymnasium Norf hat unter Berücksichtigung des Schulprogramms als überfachliche Grundsätze für die Arbeit im Unterricht beschlossen, dass die im Referenzrahmen Schulqualität NRW formulierten Kriterien und Zielsetzungen als Maßstab für die kurz- und mittelfristige Entwicklung der Schule gelten sollen. Gemäß dem Schulprogramm sollen insbesondere die Lernenden als Individuen mit jeweils besonderen Fähigkeiten, Stärken und Interessen im Mittelpunkt stehen. Die Fachgruppe vereinbart, der individuellen Kompetenzentwicklung (Referenzrahmen Kriterium 2.2.1) und den herausfordernd und kognitiv aktivierenden Lehr- und Lernprozessen (Kriterium 2.2.2) besondere Aufmerksamkeit zu widmen. Die Planung und Gestaltung des Unterrichts soll sich deshalb an der Heterogenität der Schülerschaft orientieren (Kriterium 2.6.1).

In Absprache mit der Lehrerkonferenz sowie unter Berücksichtigung des Schulprogramms hat die Fachkonferenz Informatik die folgenden fachmethodischen und fachdidaktischen Grundsätze beschlossen.

1. Die Ziele einzelner Unterrichtsstunden und der gesamten Unterrichtsreihe sind für die Schülerinnen und Schüler transparent. Ebenso ist der fachliche bzw. curriculare Zusammenhang (ggf. auch fächerübergreifend) deutlich.
2. Die Entwicklung informatischer Kompetenzen folgt konsequent dem Spiralprinzip. Programmierkonzepte, Modelle, Strategien, Fachbegriffe und wesentliche Beispiele, auf die sich die Informatiklehrkräfte verständigt haben, werden verbindlich im Fachunterricht eingeführt und bei einer vertiefenden Behandlung wieder aufgegriffen.
3. Am Verstehen orientiertes Arbeiten baut tragfähige Grundvorstellungen auf und korrigiert mögliche Fehlvorstellungen. Die Visualisierung von Fragestellungen und Lösungswegen führen zur Entwicklung eines umfassenden informatischen Verständnisses.
4. Alle Verfahren werden an hinreichend vielen Beispielen produktiv geübt.
5. Grundlegende informatische Kompetenzen auch aus weiter zurückliegenden Unterrichtsvorhaben werden regelmäßig im Unterricht wiederholt und durch vernetzte Anwendungsaufgaben gefestigt.
6. Der reflektierte und sachgerechte Einsatz von Entwicklungssoftware (z.B. Greenfoot, BlueJ) und die Nutzung von Online-Ressourcen (interaktive Tutorials, Videos) sind Gegenstand des Unterrichts.
7. Im Unterricht wird auf eine angemessene Fachsprache geachtet. Die Fachsprache wird von Lehrerinnen und Lehrern situationsangemessen korrekt benutzt. Lernende dürfen in explorativen oder kreativen Arbeitsphasen zunächst intuitive Formulierungen verwenden. In weiteren Phasen des Unterrichts werden sie dazu angehalten, die intuitiven Formulierungen zunehmend durch Fachsprache zu ersetzen.
8. Die Bedeutung der Informatik für die Lebenswirklichkeit und Lebensplanung der Schülerinnen und Schüler wird durch die Einbindung von Alltagssituationen hervorgehoben.
9. Der Informatikunterricht befähigt die Schülerinnen und Schüler dazu, geeignete Problemstellungen aus ihrem eigenen Alltag informatisch zu modellieren und unter Zuhilfenahme von eigener Software zu lösen.
10. Die Schülerinnen und Schüler erkennen zunehmend die fachübergreifende und fachverbindende Bedeutung der Informatik für die Wissenschaft und die damit verbundene Verantwortung für die Gesellschaft.
11. Binnendifferenzierung ist ein grundlegendes Prinzip im Informatikunterricht. Die Lehrkräfte setzen hierzu differenzierende Materialien und Hilfen ein (z.B. unterstützende Codekommentare, zu programmierende Zusatzfeatures), variieren die Rollen der Lernenden und nutzen kooperative Lernformen. Dabei werden sowohl fordernde als auch fördernde Aufgabenvariationen und Methoden eingesetzt. Lerntempo, Leistungsniveau und Lerntyp der Schülerinnen und Schüler finden entsprechende Berücksichtigung.
12. Ungewöhnliche Lösungsansätze werden im Unterricht angeregt und können als Gegenstand des weiteren Unterrichts aufgenommen werden. In Klassenarbeiten sind alternative Lösungswege zugelassen, dabei sind die fachliche Richtigkeit und die Lösungsqualität Kriterien zur Bewertung.

13. Materialien zum individualisierten Lernen (z. B. Arbeitsblätter, Lernvideos, Online-Kurse) unterstützen den Lernenden beim Kompetenzerwerb.
14. Die Lernenden führen über alle Jahrgänge hinweg einen Hefter, in dem im Unterricht erarbeiteten Inhalte festgehalten werden. Softwareprodukte und weitere Ergebnisse werden digital auf einem persönlichen Stick gespeichert. Über die Lernplattform Moodle werden umfangreiche Materialien zum Nacharbeiten und Üben angeboten.
15. Besonderer Wert wird auf kooperative Arbeitsformen gelegt, die nach vorgestellten Software-Entwicklungsmodellen einen dokumentierten Arbeitsprozess durchlaufen, der zu vereinbarten Deadlines zu Teilergebnissen führt. Die Entwicklung eines Softwareprojekts soll so realitätsnah simuliert werden.

4.2 Lehr- und Lernmittel

Aufgrund des steten Wandels der fachlichen Inhalte hat die Fachkonferenz Informatik beschlossen, vorläufig kein Lehrwerk anzuschaffen. Große Teile des Unterrichts orientieren sich vor allem in der Einführungsphase am Lehrwerk Informatik vom Schöningh-Verlag, der sich eng an die Abiturvorgaben hält. Dieses steht ebenso wie die weiteren Bücher der Reihe allen Schülern in der Mediothek sowohl zur Ausleihe als auch als Präsenzexemplar zur Verfügung. Darüber hinaus hält die Mediothek in Form eines Handapparats Informatik Bücher zu allen relevanten Themenbereichen der Oberstufe bereit.

Weitere Materialien (Übersichten, Aufgaben, Übungsklausuren, Referate, etc.) werden über die Lernplattform Moodle zur Verfügung gestellt, auf die jeder Schüler von zuhause wie auch aus der Schule zugreifen kann.

Es wird grundsätzlich frei zugängliche, kostenlose Software genutzt, um allen Schülern einen unkomplizierten Zugang auch von zuhause aus zu ermöglichen.

4.3 Zuordnung Medienkompetenzen

Es liegt in der Natur des Faches Informatik, dass ausnahmslos alle im Medienpass NRW aufgeführten Kompetenzen intensiv geschult werden.

5 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 aktuelle schulinterne Curriculum (siehe 2.1) ist zunächst bis 2021 für den ersten Durchgang durch die gymnasiale Oberstufe nach Erlass des Kernlehrplanes verbindlich. Erstmals nach Ende der Einführungsphase im Sommer 2019, 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. Eventuelle neue Vorgaben für das Abitur werden rechtzeitig eingearbeitet.

Nach Abschluss des Abiturs 2021 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.