

Informatik



Schulinterner Lehrplan zum Kernlehrplan für die gymnasiale Oberstufe

Inhaltsverzeichnis

1. Die Fachgruppe Informatik des Friedrich-Rückert-Gymnasiums Düsseldorf.....	3
2. Entscheidungen zum Unterricht.....	4
1.1 Übersichtsraster Unterrichtsvorhaben.....	4
1.2 Konkretisierte Unterrichtsvorhaben	8
1.3 Einführungsphase	10
Unterrichtsvorhaben EF-I: Einführung in die Informatik	11
Unterrichtsvorhaben EF-II: Einführung in die Informatik – Informatisches Problemlösen und	
Algorithmen als Beschreibung von Abläufen	13
Unterrichtsvorhaben EF-III: Einführung in die OOP mit Greenfoot	15
Unterrichtsvorhaben EF-IV: Suchen und Sortieren	18
Unterrichtsvorhaben EF-V: Softwareentwicklung	20
1.4 Qualifikationsphase.....	23
Unterrichtsvorhaben Q1-I:	24
Unterrichtsvorhaben Q1-2: Sicherheit/ Kryptographie	25
Unterrichtsvorhaben Q1-III:	28
Unterrichtsvorhaben Q1-IV:	31
Unterrichtsvorhaben Q1-IV:	35
Unterrichtsvorhaben Q1-V:	39
Unterrichtsvorhaben Q2-I:	41
Unterrichtsvorhaben Q2-II:	46
Unterrichtsvorhaben Q2-III:	48
Unterrichtsvorhaben Q2-IV:	50
1.5 Grundsätze der fachmethodischen und fachdidaktischen Arbeit.....	53
1.6 Grundsätze der Leistungsbewertung und Leistungsrückmeldung	54
Beurteilungsbereich Klausuren.....	54
Beurteilungsbereich Sonstige Mitarbeit.....	55
3. Entscheidungen zu fach- und unterrichtsübergreifenden Fragen.....	57
4. Qualitätssicherung und Evaluation	57

Die Fachkonferenz Informatik des Friedrich-Rückert-Gymnasiums in Düsseldorf.
 Redaktionell verantwortlich: Sebastian Horn (Ho) und Cornelia Walter (Wr).

Stand: 02.02.2015

1. Die Fachgruppe Informatik des Friedrich-Rückert-Gymnasiums Düsseldorf

Das Friedrich-Rückert-Gymnasium ist eines von siebzehn öffentlichen Gymnasien der Landeshauptstadt Düsseldorf. Es liegt im Norden der Stadt und hat eine bezüglich des sozialen und ethnischen Hintergrundes heterogene Schülerschaft. Das Friedrich-Rückert-Gymnasium ist in der Sekundarstufe I drei- bis vierzünftig und wird als Ganztagsgymnasium geführt.

In die Einführungsphase der Sekundarstufe II werden regelmäßig mehrere Schülerinnen und Schüler neu aufgenommen.

Der Unterricht findet im 45-Minuten-Takt statt, die Kursblockung sieht grundsätzlich für Grundkurse eine, für Leistungskurse zwei Doppelstunden vor.

Den im Schulprogramm ausgewiesenen Zielen, Schülerinnen und Schüler ihren Begabungen und Neigungen entsprechend individuell zu fördern und ihnen Orientierung für ihren weiteren Lebensweg zu bieten, fühlt sich die Fachgruppe Informatik selbstverständlich verpflichtet:

Schülerinnen und Schüler mit Übergangs- und Lernschwierigkeiten werden intensiv durch regelmäßige Sprechstunden der Lehrkräfte und dort getroffene Lernvereinbarungen unterstützt.

Im Bereich der Sekundarstufe II kooperiert unsere Schule mit dem Leibniz-Montessori-Gymnasium und bietet mit ihnen zahlreiche gemeinsame Kurse an, unter anderem GK im Fach Informatik. In der Regel gibt es bei uns einen GK in jeder Jahrgangsstufe. Informatik kann als Leistungskurs „extern“ gewählt werden, die Schüler besuchen den zentralen LK Informatik, der für alle Schüler der Landeshauptstadt angeboten.

Das Fach Informatik wird am Friedrich-Rückert-Gymnasium erst in der gymnasialen Oberstufe angeboten.

Alle Schülerinnen und Schüler und alle Lehrerinnen und Lehrer haben einen eigenen Account im schulinternen Netz und in der schulunabhängigen Lernplattform „lo-net2.de“, letztere kann auch von zu Hause genutzt werden und verzahnt das schulische Lernen mit außerschulischen Lernorten.

Der Unterricht der Sekundarstufe II wird mit Hilfe der Programmiersprache Java durchgeführt. In der Einführungsphase kommt dabei die Java-Lernumgebung Greenfoot zum Einsatz, welche einen intuitiven Zugang zur Programmierung in Java insbesondere für Neueinsteiger erleichtert.

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 Friedrich-Rückert-Gymnasiums aus zwei unterrichtenden Lehrkräften, denen drei Computerräume mit 31, 20 bzw. 17 Computerarbeitsplätzen und ein Selbstlernzentrum mit 4 Plätzen zur Verfügung stehen. Alle Arbeitsplätze sind an das schulinterne Rechnernetz angeschlossen, so dass Schülerinnen und Schüler über einen individuell gestaltbaren Zugang zum zentralen Server der Schule alle PC-Arbeitsplätze zum Zugriff auf ihre eigenen Daten, zur Recherche im Internet oder zur Bearbeitung schulischer Aufgaben verwenden können.

2. Entscheidungen zum Unterricht

2.1. Übersichtsraster Unterrichtsvorhaben

Einführungsphase	
<p><u>Unterrichtsvorhaben E-I</u></p> <p>Thema: <i>Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Informatiksysteme • Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Einzelrechner • Dateisystem • Internet • Einsatz von Informatiksystemen <p>Zeitbedarf: 6 Stunden</p>	<p><u>Unterrichtsvorhaben E-II</u></p> <p>Thema: <i>Grundlagen der objektorientierten Analyse, Modellierung und Implementierung anhand von statischen Grafikszenen</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: 8 Stunden</p>

Einführungsphase	
<p><u>Unterrichtsvorhaben E-III</u></p> <p>Thema: <i>Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen in Java anhand von einfachen Animationen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Modellieren • Implementieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Objekte und Klassen • Syntax und Semantik einer Programmiersprache • Analyse, Entwurf und Implementierung einfacher Algorithmen <p>Zeitbedarf: 18 Stunden</p>	<p><u>Unterrichtsvorhaben E-IV</u></p> <p>Thema: <i>Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand von grafischen Spielen und Simulationen</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 • Syntax und Semantik einer Programmiersprache • Analyse, Entwurf und Implementierung einfacher Algorithmen <p>Zeitbedarf: 18 Stunden</p>

Einführungsphase	
<p><u>Unterrichtsvorhaben E-V</u></p> <p>Thema: <i>Such- und Sortialgorithmen anhand kontextbezogener Beispiele</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Modellieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Algorithmen <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Algorithmen zum Suchen und Sortieren • Analyse, Entwurf und Implementierung einfacher Algorithmen <p>Zeitbedarf: 12 Stunden</p>	<p><u>Unterrichtsvorhaben E-VI</u></p> <p>Thema: <i>Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes</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"> • Informatik, Mensch und Gesellschaft • Informatiksysteme <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Wirkungen der Automatisierung • Geschichte der automatischen Datenverarbeitung • Digitalisierung <p>Zeitbedarf: 12 Stunden</p>
Summe Einführungsphase: 74	

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: 8 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: 20 Stunden</p>

2.2. Konkretisierte 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.

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. Dabei wurden ca. 75 Prozent der Bruttounterrichtszeit verplant.

Während der Fachkonferenzbeschluss zum „Übersichtsraster Unterrichtsvorhaben“ zur Gewährleistung vergleichbarer Standards sowie zur Absicherung von Kurswechslern und Lehrkraftwechseln für alle Mitglieder der Fachkonferenz Bindekraft entfalten soll, besitzt die Ausweisung „konkretisierter Unterrichtsvorhaben“ (Kapitel 2.1.2) empfehlenden Charakter. 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 2.4 zu entnehmen sind. Begründete Abweichungen bezüglich der konkretisierten Unterrichtsvorhaben sind im Rahmen der pädagogischen Freiheit der Lehrkräfte jederzeit möglich. Sicherzustellen bleibt allerdings auch hier, dass im Rahmen der Umsetzung der Unterrichtsvorhaben insgesamt alle prozess- und inhaltsbezogenen Kompetenzen des Kernlehrplans Berücksichtigung finden. Dies ist durch entsprechende Kommunikation innerhalb der Fachkonferenz zu gewährleisten.

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

In der Einführungsphase kann die didaktische Bibliothek GLOOP verwendet werden. Die folgenden Installationspakete und Dokumentationen stehen zur Verfügung:

- Installationspaket(Windows) (Download)
- Installationspaket(MacOS) (Download)
- Installationspaket(Linux) (Download)
- Dokumentationen (Download)

In der Qualifikationsphase werden die Unterrichtsvorhaben unter Berücksichtigung der Vorgaben für das Zentralabitur Informatik in NRW konkretisiert.

- Materialien zur Qualifikationsphase Q1 (Download)
- Materialien zur Qualifikationsphase Q2 (Download)

In der Einführungsphase ist das Lehrwerk „Informatik 1 – Softwareentwicklung mit Greenfoot und BlueJ“ (Schöningh Verlag) eingeführt. Nach Neuauflage (erst ab Oktober 2014) werden entsprechende Querverweise zu den jeweiligen Unterrichtsvorhaben in diesen Schulinternen Lehrplan eingefügt.

Für die Qualifikationsphase werden die Lehrwerke „Informatik 2 – Modellierung, Datenstrukturen und Algorithmen“ sowie „Informatik 3 – Netzwerkanwendungen, Informatik und Gesellschaft, Datenbanken und Theoretische Informatik“ bereits genutzt. Ferner können Aufgaben und Problemstellungen aus der Lehrwerkreihe „Informatik“ des Klett-Verlages eingebunden werden. Nach Neuauflage (erst ab Oktober 2015) werden entsprechende Querverweise zu den jeweiligen Unterrichtsvorhaben in diesen Schulinternen Lehrplan eingefügt.

Das Schulministerium stellt weitere Materialien zur Verfügung. Die Internet-Adresse zur Suche lautet:

<http://www.standardsicherung.schulministerium.nrw.de/materialdatenbank/nutzersicht/list.php?suche=INformatik&path=/0/10/>.

2.3. 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: Einführung in die Informatik

Bemerkung: Thema und Leitfragen sind vom Unterrichtsvorhaben EF-II nicht trennbar.

Thema: Einführung in die Informatik

Leitfragen: Womit beschäftigt sich das Fach Informatik in der GOST? Was sind die Fragestellungen und Meilensteine der Informatik?

Vorhabenbezogene Konkretisierung:

Das erste Unterrichtsvorhaben stellt eine allgemeine Einführung in das Fach Informatik dar. Dabei sollen die Meilensteine in der Geschichte der Informatik und grundlegende Prinzipien wie das EVA-Prinzip und die Von-Neumann-Architektur erarbeitet werden.

Mit Hilfe eines geschichtlichen Abrisses und populärer informatischer Fallbeispiele sollen die Wechselbeziehung zwischen der Informatik und der Gesellschaft thematisiert und die Auswirkungen des Einsatzes von Informatiksystemen auf die Gesellschaft deutlich werden.

Des Weiteren soll der grundlegende Aufbau eines Rechnersystems im Sinne der Von-Neumann-Architektur erarbeitet werden und mit dem grundlegenden Prinzip der Datenverarbeitung (Eingabe-Verarbeitung-Ausgabe) in Beziehung gesetzt werden.

Das gesamte Vorhaben wird durch eine eintägige Exkursion in das Heinz-Nixdorf-Forum nach Paderborn flankiert.

Zeitbedarf: 6 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
1. Geschichte und „Geschichtchen“ der Informatik und ihre Auswirkungen auf die heutige Gesellschaft	Die Schülerinnen und Schüler <ul style="list-style-type: none"> erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A) bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A) 	<ul style="list-style-type: none"> Film: „Als die Computer rechnen lernten“ Fallbeispiele (berühmte Softwarefehler: z.B. Scud-Rakete im Irak-Krieg¹, Softwarefehler bei EC-Karten², Apollo-Mondlandung³, Ariane 5)
2. Aufbau informatischer Systeme <ul style="list-style-type: none"> Identifikation typischer Arbeitsweisen und Komponenten informatischer Systeme. Herleitung der VN-Architektur Identifikation des EVA-Prinzips als Prinzip der Verarbeitung von Daten und Grundlage der VN-Architektur. 	Die Schülerinnen und Schüler <ul style="list-style-type: none"> beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der VN-Architektur (A) nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation 	<ul style="list-style-type: none"> Aus den vorherigen Beispielen und Diskussionen werden das EVA-Prinzip die in der VN-Architektur vorkommenden Komponenten herausgearbeitet. Hier ist die Vergabe von Referaten oder eine arbeitsteilige Erarbeitung angedacht.

¹ u.a. http://de.wikipedia.org/wiki/MIM-104_Patriot

² Informatik 1 – Schöningh – Seite 125

³ u.a. <http://www.bernd-leitenberger.de/apollo-ueberfluessige-fakten.shtml>

Unterrichtsvorhaben EF-II: Einführung in die Informatik – Informatisches Problemlösen und Algorithmen als Beschreibung von Abläufen

Bemerkung: Thema und Leitfragen sind vom Unterrichtsvorhaben EF-I nicht trennbar.

Thema: Einführung in die Informatik – Informatisches Problemlösen und Algorithmen als Beschreibung von Abläufen

Leitfragen: Wie beschreibt man in der Informatik Abläufe computernah? Was zeichnet Algorithmen aus und wie kann man sie vergleichen?

Vorhabenbezogene Konkretisierung:

Anhang von alltäglichen Beispielen werden zunächst der Algorithmus-Begriff und seine Eigenschaften herausgearbeitet. Dabei sollen Algorithmen sowohl umgangssprachlich wie auch im PAP dargestellt. Bedeutsam ist dabei die Herausarbeitung der grundlegenden algorithmischen Strukturen (Sequenz, bedingte Verzweigung und kopfgesteuerte Schleife) und die Überprüfung der Eigenschaften von Algorithmen.

Im Anschluss werden anhand von konkreten Algorithmen die erarbeiteten Techniken geübt, Fragestellung zu Grenzen von Algorithmen thematisiert und die Idee der rekursiv formulierten Algorithmen erarbeitet.

Beim Labyrinth-Problem führt zum ersten Mal auf Modellentscheidungen (welche Sensoren hat der Akteur? Welche Aufträge und Anfragen liegen vor?) und dient als Übergang zur ersten Implementierung in Java (Unterrichtsvorhaben EF-III).

Zeitbedarf: 9 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
1. Der Algorithmus-Begriff	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> kennen den Begriff „Algorithmus“ und seine Eigenschaften, stellen Algorithmen umgangssprachlich und grafisch (PAP) dar, analysieren und erläutern einfache Algorithmen (A) 	<ul style="list-style-type: none"> Ausgehend von den berühmten Softwarefehler⁴ und mit Hilfe alltäglicher Beispiele wird der Algorithmus-Begriff⁵ entwickelt und an Beispielen überprüft, geschärft und eingeübt. Mögliche Beispiele: Papierflieger (Sequenz), Kochrezept (Sequenz und bedingte Verzweigung) ... Material: PAPDesigner
2. Beispiele für einfache Algorithmen und ihre Darstellung	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> analysieren und erläutern einfache Algorithmen und Programme (A), entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M), modifizieren einfache Algorithmen (A), testen Programme schrittweise anhand von Beispielen (A) 	<ul style="list-style-type: none"> Euklidischer Algorithmus⁶: Darstellung eines iterativen mathematischen Algorithmus mit Hilfe eines PAPs (Bedingte Verzweigung, kopfgesteuerte Schleife) und ugs. Beschreibung Labyrinth-Problem: Die Rechte-Hand-Regel Darstellung in Umgangssprache und im PAP / Testen von Algorithmen / Übertragung auf vergleichbare Labyrinth / Grenze des Algorithmus (Problemklasse) Türme von Hanoi⁷: Herausarbeiten von Regeln und Fragestellung aus der Geschichte, Reduzierung, praktisches Ausprobieren, Darstellung als Algorithmus, Zeitproblematik (Grenze der Umsetzung des Algorithmus) Rekursive Darstellung; Eigenschaften von Rekursionen (Selbstaufruf mit Laufbedingung bei Verkleinerung des Problems)

⁴ Unterrichtsvorhaben EF-I

⁵ u.a. <http://www.netzmafia.de/skripten/buecher/perl/algorithmen.pdf> (Kapitel 2, Seite 15-21)
oder http://intranet.lsg.musin.de/sj1213/images/e/e4/Ppt_EinstiegAlgorithmen_LB.pdf (Folien 1-7)

⁶ z.B. http://de.wikipedia.org/wiki/Euklidischer_Algorithmus

⁷ Simulation und Geschichte unter <http://www.blinde-kuh.de/spiele/hanoi/>
vertiefende Materialien: <http://www.mathematik.uni-muenchen.de/~hinz/turm.pdf>

Unterrichtsvorhaben EF-III: Einführung in die OOP mit Greenfoot

Thema: Einführung in die OOP mit Greenfoot

Leitfragen: Was sind Klassen, Objekte, Methoden, Attribute? Wie interagieren Objekte unterschiedlicher und gleicher Klassen miteinander? Wie werden Algorithmen entwickelt, dargestellt und in der Programmiersprache Java umgesetzt?

Vorhabenbezogene Konkretisierung:

Dieses Unterrichtsvorhaben beinhaltet eine Einführung in die OOP. Dabei werden Algorithmen erstellt und implementiert. In Kurzprojekten werden die Klassen und ihre Beziehungen in Form von UML-Diagrammen modelliert.

Begonnen wird mit einem Greenfoot-Szenario (i.d.R. Roboter, Ameisen, Krabben o.ä.). Nach der sukzessiven Einführung der zu behandelnden Kontrollstrukturen Verzweigung, Schleifen werden zunehmend komplexere Problemstellungen von den SuS erarbeitet und gelöst, wobei die Modellierung als Vorstufe der Implementation eingefordert wird.

In einem weiteren Schritt sollen Objekte miteinander interagieren können (z.B. Roboter und Akku, zwei Roboter). Hierbei wird der Unterschied zwischen drei Beziehungsarten (Hat, Kennt und Ist) erarbeitet und visualisiert.

Zeitbedarf: 25 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p>a) Grundlagen der OOP/Algorithmik</p> <ul style="list-style-type: none"> • Erarbeitung der Bedeutung von Klassen, Objekten und Methoden • Darstellung von Klassen und Objekten in UML-Notation • Benutzung von Objekten • Erlernen grundlegender Algorithmen mit Kontrollstrukturen • Erarbeitung des Unterschieds zwischen Aufträgen und Abfragen anhand von Anwendungsbeispielen <p>b) Objektbeziehungen modellieren und umsetzen</p> <ul style="list-style-type: none"> • Analyse von Problemstellungen, bei denen die Interaktion von Objekten notwendig ist • Darstellung der Hat-, Kennt- und Ist-Beziehung in Form von Klassendiagrammen • Erarbeitung, Programmierung und Bewertung verschiedener Umsetzungsmöglichkeiten der Beziehungen 	<p>Die Schülerinnen und Schüler...</p> <ul style="list-style-type: none"> • ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M). • stellen den Zustand eines Objekts dar (D). • 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). • dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D). • analysieren und erläutern eine objektorientierte Modellierung (A) • modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M). • modellieren Klassen unter Verwendung von Vererbung (M). • stellen die Kommunikation zwischen Objekten grafisch 	<p>Greenfoot</p> <p>Szenarien für Greenfoot, z.B.:</p> <p>Wombats, Roboter, Krabben, Ameisen usw.</p> <p>PAP-Designer</p>

<p>c) Einführung des Variablenkonzepts/Datentypen</p> <ul style="list-style-type: none">• Problemorientierte Erarbeitung der Benutzung von Variablen als universellem Speicher.• Benutzung der Datentypen Integer, String, Char, Boolean. <p>d) Vertiefung der Inhalte anhand eines weiteren Anwendungsbeispiels</p>	<p>dar (M).</p> <ul style="list-style-type: none">• stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D).• analysieren und erläutern eine objektorientierte Modellierung (A).• implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I).• testen Programme schrittweise anhand von Beispielen (I).• nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K).	
---	---	--

Unterrichtsvorhaben EF-IV: Suchen und Sortieren

Thema: Suchen und Sortieren

Leitfragen: Wie funktioniert die effektive Suche nach Objekten? Wie funktioniert das Sortieren von Objekten? Welche Algorithmen gibt es und wie effizient sind diese?

Vorhabenbezogene Konkretisierung: Anhand von alltäglichen Beispielen wird das Suchen nach Objekten motiviert. Dabei wird zunächst die iterative Suche betrachtet. Es soll deutlich werden, dass ein Objekt wesentlich schneller gefunden werden kann, wenn eine Vorsortierung der zu durchsuchenden Objekte vorliegt. Daraus soll der Algorithmus der binären Suche entwickelt werden. Die benötigte Vorsortierung motiviert die Betrachtung von Sortialgorithmen. Als einfache Algorithmen wird hier beispielhaft der Selection- oder Bubblesort betrachtet. Dazu gehört die Modellierung, Implementierung in Greenfoot (in einem vorgefertigtem Szenario) und Untersuchung auf Effizienz, die sich auf die Anzahl der Vergleiche und Vertauschungen beschränkt. Die Algorithmen in diesem Unterrichtsvorhaben werden in Programmablaufplänen modelliert.

Zeitbedarf: 10 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
a) Suchalgorithmen (iterative und binäre Suche) b) Einfache Sortieralgorithmen (Selection- / Bubblesort)	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D), entwerfen einen weiteren Algorithmus zum Sortieren (M) beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeitaufwand und Speicherplatzbedarf. (A) <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> analysieren und erläutern Algorithmen und Programme (A), analysieren und erläutern einfache Algorithmen und Programme (A), entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M), implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen 	<ul style="list-style-type: none"> Suchen in Alltagsbeispielen (z.B.: CD-Regal, Suche der größten/kleinsten Zahl) Betrachtung der Effizienz der iterativen Suche Optimierung der iterativen Suche zur binären Suche. (Die Vorsortierung als Bedingung) Beschreibung der Algorithmen in PAPs Umsetzung des Bubblesorts oder Selectionsorts in Greenfoot (ohne Arrays) Zählen von Zugriffen / Vergleichen / (Dreieckstausch)

Unterrichtsvorhaben EF-V: Softwareentwicklung

Thema: Softwareentwicklung

Leitfragen: Wie wird Software entwickelt? Welche Phasen durchläuft Software während der Entwicklung? Welche Bereiche werden strikt voneinander getrennt (MVC-Konzept)?

Vorhabenbezogene Konkretisierung:

Schülerinnen und Schüler durchlaufen in diesem UV einen vollständigen Softwareentwicklungsprozess am Beispiel des Spiels „Verflixte 7“.

Im Rahmen der Unterrichtsreihe erlernen sie die einzelnen Phasen OOA , OOD bis hin zur OOP. In der OOA Phase werden beteiligte Objekte mit ihren Fähigkeiten und Eigenschaften mit Hilfe der Substantiv-Verb Methode, Use-Case Diagramme und der CRC Methode identifiziert. Die OOD dient der Konkretisierung der OOA und Implementationsdiagramme werden erstellt.

Der letzte Schritt ist die Umsetzung des Projekts/ Modells mit Java einschließlich der Gestaltung einer grafischen Oberfläche. Sequenzdiagramme helfen das System zu testen und bei Bedarf zu verbessern.

Zeitbedarf: 20 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p>a) Exkurs binäre Zahlen</p> <p>Im Rahmen eines kurzen Exkurses werden Fragestellungen rund um die Binärdarstellung besprochen.</p> <p>b) OOA</p> <p>Die OOA dient der Untersuchung der Aufgabenstellung und führt zu einem ersten UML-Modell. Die SuS benutzen dazu die Fachmethoden: USE-Case Diagramme, Substantiv Verb Methode, CRC Karten</p> <p>c) OOD</p> <p>Die OOD dient der Konkretisierung und führt zu einem Implementationsdiagramm</p> <p>d) OOP</p> <p>Mit BlueJ wird das Modell implementiert und unter anderem mit Hilfe des Debuggers und von Sequenzdiagrammen auf seine Laufbarkeit getestet.</p> <p>e) MVC Konzept</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> stellen ganze Zahlen und Zeichen in Binärcodes dar (D), interpretieren Binärcodes als Zahlen und Zeichen (D). 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), modellieren Klassen unter Verwendung von Vererbung (M), ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M), ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M), stellen den Zustand eines Objekts dar (D), stellen die Kommunikation zwischen Objekten grafisch dar (M), stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), implementieren Klassen in einer Programmiersprache (I). analysieren und erläutern einfache Algorithmen und Programme (A), entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M), implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I), nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D). implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), interpretieren Fehlermeldungen und korrigieren den Quellcode (I). dokumentieren Klassen durch Beschreibung der Funktionalität der Me- 	<ul style="list-style-type: none"> Use Case Diagramme Substantiv-Verb Methode CRC Karten Sequenzdiagramme BlueJ mit Simple Gui Extension Java Editor

<p>Das Modell wird mit Hilfe des Java-Editors um eine einfache grafische Oberfläche erweitert.</p>	<p>thoden (D).</p> <ul style="list-style-type: none">• analysieren und erläutern eine objektorientierte Modellierung (A)	
--	--	--

2.4. 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

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.

Unterrichtsvorhaben Q1-2: Sicherheit/ Kryptographie

Thema: Sicherheit? Wofür eigentlich?! Modellierung und Implementation verschiedener Verschlüsselungsmethoden und deren Analyse unter sicherheitsrelevanten Aspekten und Gemeinsamkeiten

Leitfragen: Welche Verschlüsselungsmethoden gibt es und wie sicher sind sie? Finden einer sinnvollen Modellierung um Redundanzen zu vermeiden (abstrakte Klassen). Welche Möglichkeiten bietet die Klasse String in Java? Welche Unterschiede und Gemeinsamkeiten sind mit der Datenstruktur Array vorhanden?

Vorhabenbezogene Konkretisierung:

Schülerinnen und Schüler erarbeiten in diesem UV die folgenden Verschlüsselungsmöglichkeiten:

- Caesar
- Vigenère
- Optional RSA

und untersuchen die Verfahren auf ihre Sicherheit.

Im Rahmen der Unterrichtsreihe wiederholen sie die einzelnen Phasen OOA , OOD bis hin zur OOP. Der letzte Schritt ist die Umsetzung des Projekts/ Modells mit Java einschließlich der Gestaltung einer grafischen Oberfläche. Sequenzdiagramme helfen das System zu testen und bei Bedarf zu testen und zu verbessern.

Zeitbedarf: 20 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p>a) Caesar als erstes Verschlüsselungsverfahren (ver- und Entschlüsseln)</p> <p>b) Vigenère als zweites Verschlüsselungsverfahren (ver- und Entschlüsseln)</p> <p>c) OOA</p> <p>Die OOA dient der Untersuchung der Aufgabenstellung und führt zu einem ersten UML-Modell. Die SuS benutzen dazu die Fachmethoden: Use-Case Diagramme, Substantiv Verb Methode, CRC Karten</p> <p>d) OOD</p> <p>Die OOD dient der Konkretisierung und führt zu einem Implementationsdiagramm</p> <p>e) OOP</p> <p>Mit BlueJ wird das Modell implementiert und unter anderem mit Hilfe des Debuggers und von Sequenzdiagrammen auf seine Lauffähigkeit getestet.</p> <p>f) MVC Konzept (optional)</p> <p>Das Modell wird mit Hilfe des Java-Editors um eine einfache grafische Oberfläche erweitert.</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> • Verschlüsseln einen Klartext mit Hilfe der Caesar Verschlüsselung • Entschlüsseln einen Geheimtext mit Hilfe der Caesar Verschlüsselung • Erläutern das Verfahren nach Caesar • Verschlüsseln einen Klartext mit Hilfe der Vigenère Verschlüsselung • Entschlüsseln einen Geheimtext mit Hilfe der Vigenère Verschlüsselung • Erläutern das Verfahren nach Vigenère • Untersuchen die beiden Verfahren auf Sicherheitsaspekte • Beurteilen die Sicherheit und begründen • modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), • modellieren Klassen unter Verwendung von Vererbung (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M), • ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M), • stellen den Zustand eines Objekts dar (D), • stellen die Kommunikation zwischen Objekten grafisch dar (M), • stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), 	<ul style="list-style-type: none"> • Sequenzdiagramme/ PAP • BlueJ • Java Editor

	<ul style="list-style-type: none">• implementieren Klassen in einer Programmiersprache (I).• analysieren und erläutern einfache Algorithmen und Programme (A),• entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M),• implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I),• nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D).• implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),• interpretieren Fehlermeldungen und korrigieren den Quellcode (I).• dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D).• analysieren und erläutern eine objektorientierte Modellierung (A)	
--	--	--

Unterrichtsvorhaben Q1-III:

Thema: Suchen und Sortieren auf linearen Datenstrukturen

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

Vorhabenbezogene Konkretisierung:

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

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

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

Zeitbedarf: 16 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Suchen von Daten in Listen und Arrays</p> <p>(a) Lineare Suche in Listen und in Arrays</p> <p>(b) Binäre Suche in Arrays als Beispiel für rekursives Problemlösen</p> <p>(c) Untersuchung der beiden Suchverfahren hinsichtlich ihrer Effizienz (Laufzeitverhalten, Speicherbedarf)</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> analysieren und erläutern Algorithmen und Programme (A), beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A), entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M), modifizieren Algorithmen und Programme (I), implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), 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><i>Beispiel:</i> Karteiverwaltung</p> <p>Für ein Adressverwaltungsprogramm soll eine Methode zum Suchen einer Adresse geschrieben werden.</p> <p>oder</p> <p><i>Beispiel:</i> Bundesjugendspiele</p> <p>Die Teilnehmer an Bundesjugendspielen nehmen an drei Disziplinen teil und erreichen dort Punktzahlen. Diese werden in einer Wettkampfkarte eingetragen und an das Wettkampfbüro gegeben. Zur Vereinfachung sollte sich das Modell auf die drei Disziplinen „Lauf“, „Sprung“ und „Wurf“ beschränken. Im Wettkampfbüro wird das Ergebnis erstellt. Das Programm soll dafür zunächst den Besten einer Disziplin herausuchen können und später das gesamte Ergebnis nach gewissen Kriterien sortieren können.</p> <p><i>Materialien:</i></p> <p>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.3 - Suchen und Sortieren (Download Q1-III.1)</p>
<p>2. Sortieren in Listen und Arrays - Entwicklung und Implementierung von iterativen und rekursiven Sortierverfahren</p> <p>(a) Entwicklung und Implementierung eines einfachen Sortierverfahrens für eine Liste</p>		<p><i>Beispiel:</i> Karteiverwaltung (s.o.)</p> <p>oder</p>

<ul style="list-style-type: none"> (b) Implementierung eines einfachen Sortierverfahrens für ein Feld (c) Entwicklung eines rekursiven Sortierverfahrens für ein Feld (z.B. Sortieren durch Mischen) 		<p><i>Beispiel:</i> Bundesjugendspiele (s.o.)</p> <p><i>Materialien:</i> (s.o.)</p>
<p>3. Untersuchung der Effizienz der Sortierverfahren „Sortieren durch direktes Einfügen“ und „Quicksort“ auf linearen Listen</p> <ul style="list-style-type: none"> (a) Grafische Veranschaulichung der Sortierverfahren (b) Untersuchung der Anzahl der Vergleichsoperationen und des Speicherbedarf bei beiden Sortierverfahren (c) Beurteilung der Effizienz der beiden Sortierverfahren 		<p><i>Beispiel:</i> Karteiverwaltung (s.o.)</p> <p>oder</p> <p><i>Beispiel:</i> Bundesjugendspiele (s.o.)</p> <p><i>Materialien:</i> (s.o.)</p>

Unterrichtsvorhaben Q1-IV:

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

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse Queue (a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen (b) Erarbeitung der Funktionalität der Klasse Queue (c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Queue	Die Schülerinnen und Schüler <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 Datensamm- 	<i>Beispiel:</i> Patientenwarteschlange (jeder kennt seinen Nachfolger bzw. alternativ: seinen Vorgänger) 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. Die Simulationsanwendung stellt eine GUI zur Verfügung, legt ein Wartezimmer an und steuert die Abläufe. Wesentlicher Aspekt des Projektes ist die

	<p>lungen zu (M),</p> <ul style="list-style-type: none"> • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • modifizieren Algorithmen und Programme (I), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen (I), • stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D). 	<p>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 verwaltet.</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.2 – Warteschlange (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><i>Beispiel:</i> Heftstapel In einem Heftstapel soll das Heft einer Schülerin gefunden werden.</p> <p>oder</p> <p><i>Beispiel:</i> Kisten stapeln In einem Stapel nummerierter Kisten soll eine bestimmte Kiste gefunden und an einen Kunden geliefert werden. Dazu müssen Kisten auf verschiedene Stapel gestapelt und wieder zurückgestellt werden.</p>
<p>3. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse List</p> <p>(a) Erarbeitung der Vorteile der Klasse List im Gegensatz zu den bereits bekannten linearen Strukturen</p> <p>(b) Modellierung und Implementierung einer kontextbezogenen Anwendung unter</p>		<p><i>Beispiel:</i> Abfahrtslauf Bei einem Abfahrtslauf kommen die Skifahrer nacheinander an und werden nach ihrer Zeit in eine Rangliste eingeordnet. Diese Rangliste wird in einer Anzeige ausgegeben. Ankommende Abfahrer müssen an jeder Stelle der Struktur, nicht nur am Ende oder Anfang eingefügt werden können.</p>

Verwendung der Klasse List.		<p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.2 - Listen (Download Q1-II.2)</p>
<p>4. Vertiefung - Anwendungen von Listen, Stapeln oder Schlangen in mindestens einem weiteren Kontext</p>		<p><i>Beispiel:</i> Skispringen Ein Skispringen hat folgenden Ablauf: Nach dem Sprung erhält der Springer eine Punktzahl und wird nach dieser Punktzahl in eine Rangliste eingeordnet. Die besten 30 Springer qualifizieren sich für den zweiten Durchgang. Sie starten in umgekehrter Reihenfolge gegenüber der Platzierung auf der Rangliste. Nach dem Sprung erhält der Springer wiederum eine Punktzahl und wird nach der Gesamtpunktzahl aus beiden Durchgängen in die endgültige Rangliste eingeordnet.</p> <p><i>Beispiel:</i> Terme in Postfix-Notation Die sog. UPN (<i>Umgekehrt-Polnische-Notation</i>) bzw. <i>Postfix-Notation</i> eines Terms setzt den Operator hinter die Operanden. Um einen Term aus der gewohnten Infixschreibweise in einen Term in UPN umzuwandeln oder um den Wert des Terms zu berechnen, kann ein Stack verwendet werden.</p> <p><i>Beispiel:</i> Rangierbahnhof Auf einem Güterbahnhof gibt es drei Gleise, die nur zu einer Seite offen sind. Wagons können also von einer Seite auf das Gleis fahren und nur rückwärts wieder hinausfahren. Die Wagons tragen Nummern, wobei die Nummer jedoch erst eingesehen werden kann, wenn der Wagon der vorderste an der offenen Gleisseite ist. (Zwischen den Wagons herumzurnen, um die anderen Wagonnummern zu lesen,</p>

		<p>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><i>Beispiel:</i> Autos an einer Ampel zur Zufahrtsregelung 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><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1-II.3 – Anwendungen für lineare Datenstrukturen (Download Q1-II.3)</p>
--	--	--

Unterrichtsvorhaben Q1-IV:

Thema: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

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

Vorhabenbezogene Konkretisierung:

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

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

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

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

Zeitbedarf: 20 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Nutzung von relationalen Datenbanken</p> <p>(a) <i>Aufbau von Datenbanken und Grundbegriffe</i></p> <ul style="list-style-type: none"> Entwicklung von Fragestellungen zur vorhandenen Datenbank Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, Datenbankschema <p>(b) <i>SQL-Abfragen</i></p> <ul style="list-style-type: none"> Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL (SELECT (DISTINCT) ...FROM, WHERE, AND, OR, NOT) auf einer Tabelle Analyse und Erarbeitung von SQL-Abfragen auf einer und mehrerer Tabelle zur Beantwortung der Fragestellungen (JOIN, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, <>, >, <, >=, <=, LIKE, BETWEEN, IN, IS NULL) <p>(c) <i>Vertiefung an einem weiteren Datenbankbeispiel</i></p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A), analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A), analysieren und erläutern eine Datenbankmodellierung (A), erläutern die Eigenschaften normalisierter Datenbankschemata (A), bestimmen Primär- und Sekundärschlüssel (M), ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M), modifizieren eine Datenbankmodellierung (M), modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M), bestimmen Primär- und Sekundärschlüssel (M), überführen Datenbankschemata in vorgegebene Normalformen (M), verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I), ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D), 	<p><i>Beispiel: VideoCenter</i></p> <p>VideoCenter ist die Simulation einer Online-Videothek für den Informatik-Unterricht mit Webfrontends zur Verwaltung der Kunden, der Videos und der Ausleihe. Außerdem ist es möglich direkt SQL-Abfragen einzugeben. Es ist auch möglich, die Datenbank herunter zu laden und lokal zu installieren. Unter http://dokumentation.videocenter.schule.de/old/video/index.html (abgerufen: 30. 03. 2014) findet man den Link zu dem VideoCenter-System sowie nähere Informationen. Lesenswert ist auch die dort verlinkte „Dokumentation der Fallstudie“ mit didaktischem Material, welches alternativ bzw. ergänzend zu der im Folgenden beschriebenen Durchführung verwendet werden kann.</p> <p><i>Beispiel: Schulbuchausleihe</i></p> <p>Unter www.brd.nrw.de/lerntreffs/informatik/structure/material/sek2/datenbanken.php (abgerufen: 30. 03. 2014) wird eine Datenbank zur Verfügung gestellt, die Daten einer Schulbuch-Ausleihe enthält (über 1000 Entleiher, 200 Bücher mit mehreren tausend Exemplaren und viele Ausleihvorgänge). Die Datenbank kann in OpenOffice eingebunden werden.</p>

2. Modellierung von relationalen Datenbanken

(a) Entity-Relationship-Diagramm

- Ermittlung von Entitäten, zugehörigen Attributen, Relationen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms

- Erläuterung und Modifizierung einer Datenbankmodellierung

(b) Entwicklung einer Datenbank aus einem Datenbankentwurf

- Modellierung eines relationalen Datenbankschemas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln

(c) Redundanz, Konsistenz und Normalformen

- 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)

- stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D),
- überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D).

Beispiel: Fahrradverleih

Der Fahrradverleih *BTR (BikesToRent)* verleiht unterschiedliche Typen von Fahrrädern diverser Firmen an seine Kunden. Die Kunden sind bei *BTR* registriert (Name, Adresse, Telefon). *BTR* kennt von den Fahrradfirmen den Namen und die Telefonnummer. Kunden von *BTR* können CityBikes, Treckingräder und Mountainbikes ausleihen.

Beispiel: Reederei

Die Datenverwaltung einer Reederei soll in einem Datenbanksystem umgesetzt werden. Ausgehend von der Modellierung soll mit Hilfe eines ER-Modells und eines Datenbankschemas dieser erste Entwurf normalisiert und in einem Datenbanksystem umgesetzt werden. Es schließen sich diverse SQL-Abfragen an, wobei auf die Relationenalgebra eingegangen wird.

Beispiel: Buchungssystem

In dem Online-Buchungssystem einer Schule können die Lehrer Medienräume, Beamer, Laptops, Kameras, usw. für einen bestimmten Zeitpunkt buchen, der durch Datum und die Schulstunde festgelegt ist.

Dazu ist die Datenbank zu modellieren, ggf. zu normalisieren und im Datenbanksystem umzusetzen. Weiter sollen sinnvolle Abfragen entwickelt werden.

Unter <http://mrbs.sourceforge.net> (abgerufen: 30.03. 2014) findet man ein freies Online-Buchungssystem inklusive Demo, an Hand derer man erläutern kann, worum es in dem Projekt

		<p>geht.</p> <p><i>Beispiel: Schulverwaltung</i></p> <p>In einer Software werden die Schulhalbjahre, Jahrgangsstufen, Kurse, Klassen, Schüler, Lehrer und Noten einer Schule verwaltet. Man kann dann ablesen, dass z.B. Schüler X von Lehrer Y im 2. Halbjahr des Schuljahrs 2011/2012 in der Jahrgangsstufe 9 im Differenzierungsbereich im Fach Informatik die Note „sehr gut“ erhalten hat. Dazu ist die Datenbank zu modellieren, ggf. zu normalisieren und im Datenbanksystem umzusetzen. Weiter sollen sinnvolle Abfragen entwickelt werden und das Thema Datenschutz besprochen werden.</p>
--	--	---

Unterrichtsvorhaben Q1-V:

Thema: Sicherheit und Datenschutz in Netzstrukturen

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

Vorhabenbezogene Konkretisierung:

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

Zeitbedarf: 10 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Daten in Netzwerken und Sicherheitsaspekte in Netzen sowie beim Zugriff auf Datenbanken</p> <p>(a) <i>Beschreibung eines Datenbankzugriffs im Netz anhand eines Anwendungskontextes und einer Client-Server-Struktur zur Klärung der Funktionsweise eines Datenbankzugriffs</i></p> <p>(b) <i>Netztopologien als Grundlage von Client-Server-Strukturen und TCP/IP-Schichtenmodell als Beispiel für eine Paketübermittlung in einem Netz</i></p> <p>(c) <i>Vertraulichkeit, Integrität, Authentizität in Netzwerken sowie symmetrische und asymmetrische kryptografische Verfahren (Cäsar-, Vigenère-, RSA-Verfahren) als Methoden Daten im Netz verschlüsselt zu übertragen</i></p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • beschreiben und erläutern Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A), • analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A), • untersuchen und bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informationssystemen, die Sicherheit von Informationssystemen sowie die Einhaltung der Datenschutzbestimmungen und des Urheberrechts (A), • untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informationssystemen ergeben, hinsichtlich rechtlicher Vorgaben, 	<p><i>Materialien:</i></p> <p>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben, Verschlüsselung Q1.5 - Zugriff auf Daten in Netzwerken (Download Q1-V.1)</p>

	ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A),	
2. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht	<ul style="list-style-type: none">• nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D).	<i>Materialien:</i> <i>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1 5 - Datenschutz beim Videocenter, Materialblatt-Datenschutzgesetz (Download Q1-V.2)</i>

Unterrichtsvorhaben Q2-I:

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

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

Vorhabenbezogene Konkretisierung:

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

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

Eine Tiefensuche wird verwendet, um einen in der Baumstruktur gespeicherten Inhalt zu suchen.

Zu einer Problemstellung in einem entsprechenden Anwendungskontext werden die Operationen der Datenstruktur Suchbaum thematisiert und unter der Verwendung der Klasse `BinarySearchTree` (der Materialien für das Zentralabitur in NRW) weitere Klassen oder Methoden in diesem Anwendungskontext modelliert und implementiert. Auch in diesem Kontext werden grafische Darstellungen der Bäume verwendet.

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

Zeitbedarf: 24 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Analyse von Baumstrukturen in verschiedenen Kontexten</p> <p>(a) Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit)</p> <p>(b) Aufbau und Darstellung von binären Bäumen anhand von Baumstrukturen in verschiedenen Kontexten</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), • analysieren und erläutern Algorithmen und Programme (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nicht-lineare 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), 	<p><i>Beispiel:</i> Termbaum Der Aufbau von Termen wird mit Hilfe von binären Baumstrukturen verdeutlicht.</p> <p>oder</p> <p><i>Beispiel:</i> Ahnenbaum Die binäre Baumstruktur ergibt sich daraus, dass jede Person genau einen Vater und eine Mutter hat.</p> <p><i>Weitere Beispiele für Anwendungskontexte für binäre Bäume:</i></p> <p><i>Beispiel:</i> Suchbäume (zur sortierten Speicherung von Daten) 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><i>Beispiel:</i> Entscheidungsbäume 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>

	<ul style="list-style-type: none"> • modifizieren Algorithmen und Programme (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen (I), • stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D). 	<p><i>Beispiel:</i> 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 Buchstaben.</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Binärbaum (Download Q2-I.1)</p>
<p>2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse <i>BinaryTree</i></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms</p> <p>(c) Erarbeitung der Klasse <i>BinaryTree</i> und beispielhafte Anwendung der Operationen</p> <p>(d) Implementierung der Anwendung oder von Teilen der Anwendung</p>		<p><i>Beispiel:</i> Informatikerbaum als binärer Baum</p> <p>In einem <i>binären Baum</i> werden die Namen und die Geburtsdaten von Informatikern lexikographisch geordnet abgespeichert. Alle Namen, die nach dieser Ordnung vor dem Namen im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem Namen im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.)</p> <p>Folgende Funktionalitäten werden benötigt:</p> <ul style="list-style-type: none"> • Einfügen der Informatiker-Daten in den Baum • Suchen nach einem Informatiker über den Schlüssel Name

<p>(e) <i>Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf</i></p>		<ul style="list-style-type: none"> • Ausgabe des kompletten Datenbestands in nach Namen sortierter Reihenfolge <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Binärbaum (Download Q2-I.2)</p>
<p>3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse <code>BinarySearchTree</code></p> <p>(a) <i>Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</i></p> <p>(b) <i>Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramm, grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften</i></p> <p>(c) <i>Erarbeitung der Klasse <code>BinarySearchTree</code> und Einführung des Interface <code>Item</code> zur Realisierung einer geeigneten Ordnungsrelation</i></p> <p>(d) <i>Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums</i></p>		<p><i>Beispiel: Informatikerbaum als Suchbaum</i> In einem binären <i>Suchbaum</i> werden die Namen und die Geburtsdaten von Informatikern lexikographisch geordnet abgespeichert. Alle Namen, die nach dieser Ordnung vor dem Namen im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem Namen im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.)</p> <p>Folgende Funktionalitäten werden 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><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Binärer Suchbaum (Download Q2-I.3)</p>
<p>4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäu-</p>		<p><i>Beispiel: Codierungsbäume (s.o.) oder Huffman-Codierung</i></p>

<p>men anhand weiterer Problemstellungen</p>		<p><i>oder</i></p> <p><i>Beispiel:</i> Buchindex Es soll eine Anwendung entwickelt werden, die anhand von Stichworten und zugehörigen Seitenzahlen ein Stichwortregister erstellt. Da die Stichwörter bei der Analyse des Buches häufig gesucht werden müssen, werden sie in der Klasse Buchindex als Suchbaum (Objekt der Klasse BinarySearchTree) verwaltet. 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><i>oder</i></p> <p><i>Beispiel:</i> Entscheidungsbäume (s.o.)</p> <p><i>oder</i></p> <p><i>Beispiel:</i> Termbaum (s.o.)</p> <p><i>oder</i></p> <p><i>Beispiel:</i> Ahnenbaum (s.o.)</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Anwendung Binärbaum (Download Q2-I.4)</p>
---	--	--

Unterrichtsvorhaben Q2-II:

Thema: Endliche Automaten und formale Sprachen

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

Vorhabenbezogene Konkretisierung:

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

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

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

Zeitbedarf: 20 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
1. Endliche Automaten (a) Vom Automaten in den Schülerinnen und Schülern bekannten Kontexte 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 einschließlich ihres Verhaltens auf bestimmte Eingaben (A), analysieren und erläutern Grammatiken regulärer Sprachen (A), zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A), ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A), entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M), entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M), entwickeln zur akzeptierten Sprache eines Automaten die zugehörige Grammatik (M), entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M), modifizieren Grammatiken regulärer Sprachen (M), entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M), stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D), ermitteln die Sprache, die ein endlicher Automat akzeptiert (D). beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D). 	<i>Beispiele:</i> Cola-Automat, Geldspielautomat, Roboter, Zustandsänderung eines Objekts „Auto“, Akzeptor für bestimmte Zahlen, Akzeptor für Teilwörter in längeren Zeichenketten, Akzeptor für Terme <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.2 – Endliche Automaten, Formale Sprachen (Download Q2-II.1)
2. Untersuchung und Entwicklung von Grammatiken 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 werden (d) Entwicklung regulärer Grammatiken zu endlichen Automaten		<i>Beispiele:</i> reguläre Grammatik für Wörter mit ungerader Parität, Grammatik für Wörter, die bestimmte Zahlen repräsentieren, Satzgliederungsgrammatik <i>Materialien: (s.o.)</i>
3. Grenzen endlicher Automaten		<i>Beispiele:</i> Klammerausdrücke, $a^n b^n$ im Vergleich zu $(ab)^n$

Unterrichtsvorhaben Q2-III:

Thema: Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

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

Vorhabenbezogene Konkretisierung:

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

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: 12 Stunden

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 <ul style="list-style-type: none"> • erläutern die Ausführung eines einfachen maschinennahen Programms sowie die Datenspeicherung auf einer „Von-Neumann-Architektur“ (A), • untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A). 	<i>Beispiel:</i> Addition von 4 zu einer eingegeben Zahl mit einem Rechnermodell <i>Materialien:</i> 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		<i>Beispiel:</i> Halteproblem <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 - Halteproblem (Download Q2-III.2)

Unterrichtsvorhaben Q2-IV:

Wiederholung und Vertiefung ausgewählter Kompetenzen und Inhalte des ersten Jahrs der Qualifikationsphase

Unterrichtsvorhaben EF-I

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

Leitfragen: *Womit beschäftigt sich die Wissenschaft der Informatik? Wie kann die in der Schule vorhandene informatische Ausstattung genutzt werden?*

Vorhabenbezogene Konkretisierung:

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

Zunächst wird auf den Begriff der Information eingegangen und die Möglichkeit der Kodierung in Form von Daten thematisiert. Anschließend wird auf die Übertragung von Daten im Sinne des Sender-Empfänger-Modells eingegangen. Dabei wird eine überblickartige Vorstellung der Kommunikation von Rechnern in Netzwerken erarbeitet.

Des Weiteren soll der grundlegende Aufbau eines Rechnersystems im Sinne der Von-Neumann-Architektur erarbeitet werden und mit dem grundlegenden Prinzip der Datenverarbeitung (Eingabe-Verarbeitung-Ausgabe) in Beziehung gesetzt werden.

Bei der Beschäftigung mit Datenkodierung, Datenübermittlung und Datenverarbeitung ist jeweils ein Bezug zur konkreten Nutzung der informatischen Ausstattung der Schule herzustellen. So wird in die verantwortungsvolle Nutzung dieser Systeme eingeführt.

Zeitbedarf: 6 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Information, deren Kodierung und Speicherung (a) Informatik als Wissenschaft der Verarbeitung von Informationen (b) Darstellung von Informationen in Schrift, Bild und Ton (c) Speichern von Daten mit informatischen Systemen am Beispiel der Schulrechner (d) Vereinbarung von Richtlinien zur Da-	Die Schülerinnen und Schüler <ul style="list-style-type: none"> • beschreiben und erläutern den Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A), • nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D), • nutzen das Internet zur Recherche, zum Da- 	<i>Beispiel:</i> Textkodierung Kodierung und Dekodierung von Texten mit unbekannten Zeichensätzen (z.B. Wingdings) <i>Beispiel:</i> Bildkodierung Kodierung von Bildinformationen in Raster- und Vektorgrafiken

<p>tenspeicherung auf den Schulrechnern (z.B. Ordnerstruktur, Dateibezeichner usw.)</p>	<p>tenaustausch und zur Kommunikation (K).</p>	
<p>2. Informations- und Datenübermittlung in Netzen</p> <p>(a) „Sender-Empfänger-Modell“ und seine Bedeutung für die Eindeutigkeit von Kommunikation</p> <p>(b) Informatische Kommunikation in Rechnernetzen am Beispiel des Schulnetzwerks (z.B. Benutzeranmeldung, Netzwerkordner, Zugriffsrechte, Client-Server)</p> <p>(c) Grundlagen der technischen Umsetzung von Rechnerkommunikation am Beispiel des Internets (z.B. Netzwerkadresse, Paketvermittlung, Protokoll)</p> <p>(d) Richtlinien zum verantwortungsvollen Umgang mit dem Internet</p>		<p><i>Beispiel:</i> Rollenspiel zur Paketvermittlung im Internet</p> <p>Schülerinnen und Schüler übernehmen die Rollen von Clients und Routern. Sie schicken spielerisch Informationen auf Karten von einem Schüler-Client zum anderen. Jede Schülerin und jeder Schüler hat eine Adresse, jeder Router darüber hinaus eine Routingtabelle. Mit Hilfe der Tabelle und einem Würfel wird entschieden, wie ein Paket weiter vermittelt wird.</p>
<p>3. Aufbau informatischer Systeme</p> <p>(a) Identifikation typischer Komponenten informatischer Systeme und anschließende Beschränkung auf das Wesentliche, Herleitung der „Von-Neumann-Architektur“</p> <p>(b) Identifikation des EVA-Prinzips (Eingabe-Verarbeitung-Ausgabe) als Prinzip der Verarbeitung von Daten und Grundlage der „Von-Neumann-Architektur“</p>		<p><i>Material:</i> Demonstrationshardware</p> <p>Durch Demontage eines Demonstrationsrechners entdecken Schülerinnen und Schüler die verschiedenen Hardwarekomponenten eines Informatiksystems. Als Demonstrationsrechner bietet sich ein ausrangierter Schulrechner an. Alternativ kann der Der Knowhow - Computer oder die Software MOPS zur Modellierung und Darstellung genutzt werden.</p>

2.5. Grundsätze der fachmethodischen und fachdidaktischen Arbeit

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

Überfachliche Grundsätze:

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

Fachliche Grundsätze:

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

2.6. Grundsätze der Leistungsbewertung und Leistungsrückmeldung

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

Beurteilungsbereich Klausuren

Verbindliche Absprachen:

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

Instrumente:

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

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

Kriterien

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

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

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

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

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 Übung dauern ca. 20 Minuten und umfassen den Stoff der letzten ca. 4–6 Stunden.
- Bearbeitung von schriftlichen Aufgaben im Unterricht

Kriterien

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

Die Bewertungskriterien stützen sich auf

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

Besonderes Augenmerk ist dabei auf

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

Bei Gruppenarbeiten auch auf

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

Bei Projektarbeit darüber hinaus auf

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

Grundsätze der Leistungsrückmeldung und Beratung

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

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

Die Leistungsrückmeldung kann

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

erfolgen.

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

3. Entscheidungen zu fach- und unterrichtsübergreifenden Fragen

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

Zusammenarbeit mit anderen Fächern

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

Vorbereitung auf die Erstellung der Facharbeit

Möglichst schon 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 ist gewünscht, dass Facharbeiten vergeben werden, die mit der eigenständigen Entwicklung eines Softwareproduktes verbunden sind.

4. Qualitätssicherung und Evaluation

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

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

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

Die Fachkonferenz Informatik des Friedrich-Rückert-Gymnasiums in Düsseldorf.

Redaktionell verantwortlich: Sebastian Horn (Ho) und Cornelia Walter (Wr).

Stand: 02.02.2015