

**Nicolaus-Cusanus Gymnasium,
Bergisch Gladbach**

**Schulinterner Lehrplan
Informatik**

(Stand:10.03.2021, HAAS)

Inhalt

	Seite
1 Die Fachgruppe Informatik des NCG	3
2 Entscheidungen zum Unterricht	5
2.1 Unterrichtsvorhaben	5
2.1.1 <i>Übersichtsraster Unterrichtsvorhaben</i>	6
2.1.2 <i>Konkretisierte Unterrichtsvorhaben</i>	13
2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit	52
2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung	53
2.3.1 <i>Beurteilungsbereich Klausuren</i>	53
2.3.2 <i>Beurteilungsbereich Sonstige Mitarbeit</i>	54
2.4 Lehr- und Lernmittel	57
3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen	58
4 Qualitätssicherung und Evaluation	59

1 Die Fachgruppe Informatik des NCG

Das Fach Informatik wird am NCG ab der Jahrgangsstufe 8 im Wahlpflichtbereich II (WP II) zweiständig. In der zweijährigen Laufzeit dieser Kurse wird in altersstufengerechter Weise unter anderem auf Grundlagen der Algorithmik am Beispiel einer didaktischen Lernumgebung, auf die technische Informatik am Beispiel von Schaltwerken und Schaltnetzen und auf Robotik eingegangen. werden. Darüber hinaus erhalten die Schülerinnen und Schüler dieses Differenzierungskurses einen tiefergehenden Einblick in Tabellenkalkulationen sowie die Gestaltung und Programmierung von Internetseiten (HTML, CSS).

In der Jahrgangsstufe 7 wird epochal ein für alle verpflichtender Kurs zum Umgang mit informatischen Systemen durchgeführt. Hierbei werden neben Office-Anwendungen auch erste Grundlagen der Algorithmik vermittelt.

In der Sekundarstufe II bietet das NCG für die Schülerinnen und Schüler in allen Jahrgangsstufen jeweils einen Grundkurs in Informatik an. Zukünftig sollen potentiell auch Leistungskurse in Informatik angeboten werden.

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

Der Unterricht der Sekundarstufe II wird mit Hilfe der Programmiersprache Java durchgeführt. Dabei kommt zusätzlich eine didaktische reduzierte Entwicklungsumgebung zum Einsatz, welche das Erstellen von Programmen 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.

Darüber hinaus trägt er zu einer breitgefächerten Allgemeinbildung bei, bietet gleichzeitig Raum für individuelle Spezialisierungen und ermöglicht verantwortungsvolles Handeln in einer sich schnell wandelnden und von technischen Fortschritten geprägten Welt.

Zurzeit besteht die Fachschaft Informatik des NCG aus drei Lehrkräften, denen ein Informatikraum mit 24 Computerarbeitsplätzen und ein Computerraum mit 19 Computerarbeitsplätzen zur Verfügung steht. Alle Arbeitsplätze sind an das schulinterne Rechnernetz angeschlossen, so dass Schülerinnen und Schüler über einen individuell gestaltbaren Zugang zum zentralen Server der Schule alle Arbeitsplätze der drei Räume zum Zugriff auf ihre eigenen Daten, zur Recherche im Internet oder zur Bearbeitung schulischer Aufgaben verwenden können.

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

2 Entscheidungen zum Unterricht

2.1 Unterrichtsvorhaben

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

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

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

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

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

2.1.1 Übersichtsraster Unterrichtsvorhaben

I) Einführungsphase

Im Folgenden sollen Unterrichtsvorhaben für das Fach Informatik dargestellt werden. Alle hier aufgeführten Vorhaben beziehen sich auf Grundkurse in der Einführungsphase.

Zu jedem Unterrichtsvorhaben ist eine Anknüpfung an den Kernlehrplan Informatik in Form von Kompetenzbezügen gegeben. Die aufgeführten Kompetenzen sind dabei so zu verstehen, dass das entsprechende Unterrichtsvorhaben zum Erwerb derselben beiträgt. Kompetenzerwerb ist ein kontinuierlicher und kumulativer Prozess, der sich über längere Zeiträume hinzieht und die wiederholte Beschäftigung mit entsprechenden fachlichen Gegenständen und Themen in variierenden Anwendungssituationen oder auf zunehmenden Anforderungsniveaus voraussetzt. Es kann daher nicht der Anspruch erhoben werden, dass die aufgeführten Kompetenzen nach Abschluss lediglich eines Unterrichtsvorhabens vollständig erworben wurden.

Einführungsphase	
<p><u><i>Unterrichtsvorhaben E-I</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><i>Unterrichtsvorhaben E-II</i></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

Unterrichtsvorhaben E-III

Thema:

Grundlagen der Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand von grafischen Spielen und Simulationen mit LeJOS

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Implementieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Zeitbedarf: 18 Stunden

Unterrichtsvorhaben E-IV

Thema:

Implementierung einfacher Algorithmen und GUI-Entwurf

Zentrale Kompetenzen:

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

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Zeitbedarf: 18 Stunden

Einführungsphase	
<p><u>Unterrichtsvorhaben E-V</u></p> <p>Thema: <i>Entwurf und Implementierung einfacher Sortierverfahren und eines klassischen kryptographischen Verfahrens</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Algorithmen • Daten und ihre Strukturierung • Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Algorithmen zum Suchen und Sortieren • Analyse, Entwurf und Implementierung einfacher Algorithmen • Syntax und Semantik einer Programmiersprache <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: 15 Stunden</p>
Summe Einführungsphase: 77	

II) Qualifikationsphase (Q1 und Q2) – Grundkurs / **Leistungskurs**

Qualifikationsphase 1	
<p><u>Unterrichtsvorhaben Q1-I</u></p> <p>Thema: <i>Rekursive Algorithmen und Backtracking in Anwendungskontexten</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none">• Argumentieren• Modellieren• Implementieren• Darstellen und Interpretieren• Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none">• Algorithmen• Formale Sprachen und Automaten• Informatiksysteme• Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none">• Analyse, Entwurf und Implementierung von Algorithmen• Algorithmen in ausgewählten informatischen Kontexten• Syntax und Semantik einer Programmiersprache• Nutzung von Informatiksystemen• Grenzen der Automatisierung <p>Zeitbedarf: 20 / 25 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• Informatiksysteme <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• Nutzung von Informatiksystemen <p>Zeitbedarf: 20 / 30 Stunden</p>

Qualifikationsphase 1	
<p><u>Unterrichtsvorhaben Q1-III</u></p> <p>Thema: <i>Modellierung und Implementierung dynamischer nichtlinearer Datenstrukturen am Beispiel der Binärbäume</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 ihrer 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 • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache • Nutzung von Informatiksystemen <p>Zeitbedarf: 20 / 25 Stunden</p>	<p><u>Unterrichtsvorhaben Q1-IV</u></p> <p>Thema: <i>Sicherheit und Datenschutz in Netzstrukturen und Modellierung und Implementierung von Client-Server-Anwendungen</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 • Informatik, Mensch und Gesellschaft <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 • Einzelrechner und Rechnernetzwerke • Sicherheit • Nutzung von Informatiksystemen • Wirkung der Automatisierung <p>Zeitbedarf: 15 / 20 Stunden</p>
<p>Summe Qualifikationsphase 1: 80 (105) Stunden</p>	

Qualifikationsphase 2

Unterrichtsvorhaben Q2-I / Q1-V

Thema:

Modellierung und Nutzung relationaler Datenbanken in Anwendungskontexten

Zentrale Kompetenzen:

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

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

- Datenbanken
- Algorithmen in ausgewählten informatischen Kontexten
- Syntax und Semantik einer Programmiersprache
- Nutzung von Informatiksystemen
- Sicherheit
- Wirkung der Automatisierung

Zeitbedarf: 25 / 30 Stunden

Unterrichtsvorhaben Q2-II

Thema:

Endliche Automaten und formale Sprachen

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Endliche Automaten und formale Sprachen
- **Daten und ihre Strukturierung**
- **Algorithmen**

Inhaltliche Schwerpunkte:

- Endliche Automaten (**und Kellerautomaten**)
- Grammatiken regulärer (**und kontextfreier**) Sprachen
- **Scanner, Parser und Interpreter für eine reguläre Sprache**
- Möglichkeiten und Grenzen von Automaten und formalen Sprachen
- Nutzung von Informatiksystemen

Zeitbedarf: 25 / 40 Stunden

Qualifikationsphase 2

Unterrichtsvorhaben Q2-III

Thema:

*Prinzipielle Arbeitsweise eines Computers
und Grenzen der Automatisierbarkeit*

Zentrale Kompetenzen:

- Argumentieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Einzelrechner und Rechnernetzwerke
- Grenzen der Automatisierung

Zeitbedarf: 10 Stunden

Unterrichtsvorhaben Q2-IV

Thema:

Modellierung und Implementierung dynamischer Datenstrukturen am Beispiel der Graphen

Zentrale Kompetenzen:

- Argumentieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Einzelrechner und Rechnernetzwerke
- Grenzen der Automatisierung

Zeitbedarf: 15 Stunden

Summe Qualifikationsphase 2: 60 (95) Stunden

2.1.2 Konkretisierte Unterrichtsvorhaben

Hinweis: Thema, Inhaltfelder, inhaltliche Schwerpunkte, Kompetenzen und Absprachen zur Vorhabenbezogenen Konkretisierung hat die Fachkonferenz verbindlich vereinbart. In allen anderen Bereichen (Unterrichtssequenzen und verwendeten Beispiele, Medien und Materialien) sind Abweichungen von den vorgeschlagenen Vorgehensweisen möglich. Darüber hinaus enthält dieser schulinterne Lehrplan in den Kapiteln 2.2 bis 2.3 übergreifende sowie z. T. auch jahrgangsbezogene Absprachen zur fachmethodischen und fachdidaktischen Arbeit, zur Leistungsbewertung und zur Leistungsrückmeldung. Je nach internem Steuerungsbedarf können solche Absprachen auch Vorhabenbezogen vorgenommen werden.

1) Einführungsphase

Im Folgenden sollen die in Abschnitt 1 aufgeführten Unterrichtsvorhaben konkretisiert werden. Diese Konkretisierung hat vorschlagenden Charakter, ohne die pädagogische Freiheit des Lehrenden einschränken zu wollen.

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

Die Schülerinnen und Schüler

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

Unterrichtsvorhaben EF-I

Thema: Einführung in die Nutzung von Informationssystemen 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
<p>1. Allgemeine Einführung</p> <ul style="list-style-type: none"> • Übersicht über das Fach • Einführung in die Informatiksysteme der Schule insbesondere der Lernplattform • Anleitung zur Installation zusätzlicher Software auf dem eigenen Informatiksystem 	<ul style="list-style-type: none"> • Die Schülerinnen und Schüler nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D), • nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K), • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K). 	<p>Teams</p>
<p>2. Grundlagen von Informatiksystemen und Überblick über die Geschichte der Informatik</p> <ul style="list-style-type: none"> • Darstellung von Zahlen im Binärsystem • von-Neumann-Architektur • Geschichte der Datenverarbeitung 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • Stellen ganze Zahlen und Zeichen in Binär-codes dar (D), • interpretieren Binär-codes als Zahlen und Zeichen (D), • beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „von-Neumann-Architektur“ (A). • erläutern wesentliche Grundlagen der digitalen Datenverarbeitung (A). 	

Unterrichtsvorhaben EF-II

Thema: Grundlagen der objektorientierten Analyse, Modellierung und Implementierung anhand von statischen Grafikszenen

Leitfrage: *Wie lassen sich Gegenstandsbereiche informatisch modellieren und im Sinne einer Simulation informatisch realisieren?*

Vorhabenbezogene Konkretisierung:

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

Dazu werden zunächst konkrete Gegenstandsbereiche aus der Lebenswelt der Schülerinnen und Schüler analysiert und im Sinne des objektorientierten Paradigmas strukturiert. Dabei werden die grundlegenden Begriffe der objektorientierung und Modellierungswerkzeuge wie Objektkarten, Klassenkarten oder Beziehungsdiagramme eingeführt.

Im Anschluss wird mit der Realisierung erster Projekte mit Hilfe der didaktischen Programmierumgebung BlueJ begonnen. Zur Modellierung einfacher Grafiken werden verschiedene Klassen zur Darstellung geometrischer Objekte vorgegeben. Diese werden von Schülerinnen und Schülern in Teilen analysiert und entsprechende Objekte anhand einfacher Problemstellungen erprobt. Dazu muss der grundlegende Aufbau einer Java-Klasse thematisiert und zwischen Deklaration, Initialisierung und Methodenaufrufen unterschieden werden.

Da bei der Umsetzung dieser ersten Projekte konsequent auf die Verwendung von Kontrollstrukturen verzichtet wird und der Quellcode aus einer rein linearen Sequenz besteht, ist auf diese Weise eine Fokussierung auf die Grundlagen der objektorientierung möglich, ohne dass algorithmische Probleme ablenken. Natürlich kann die Arbeit an diesen Projekten unmittelbar zum nächsten Unterrichtsvorhaben führen. Dort stehen unter anderem Kontrollstrukturen im Mittelpunkt.

Zeitbedarf: 8 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Identifikation von Objekten</p> <ul style="list-style-type: none"> • Am Beispiel eines lebensweltlichen Beispiels werden Objekte im Sinne der objektorientierten Modellierung eingeführt. • Objekte werden mit Objektkarten visualisiert und mit sinnvollen Attributen und „Fähigkeiten“, d.h. Methoden versehen. • Manche Objekte sind prinzipiell typgleich und werden so zu einer Objektsorte bzw. Objekt-klasse zusammengefasst. • Vertiefung: Modellierung weiterer Beispiele ähnlichen Musters 	<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), • modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), • stellen die Kommunikation zwischen Objekten grafisch dar (M), • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), • stellen den Zustand eines Objekts dar (D). 	<p>Arbeitsblätter, BlueJ, Rahmenprogramme</p>
<p>2. Analyse von Klassen didaktischer Lernumgebungen</p> <ul style="list-style-type: none"> • Objektorientierte Programmierung als modularisiertes Vorgehen (Entwicklung von Problemlösungen auf Grundlage vorhandener Klassen) • Aufbau einer grafischen Szene in BlueJ mithilfe der Bordmöglichkeiten von BlueJ • Analyse der Klassen des Figurenprojekts 		
<p>3. Implementierung statischer Szenen</p> <ul style="list-style-type: none"> • Grundaufbau einer Java-Klasse • Deklaration und Initialisierung von Objekten • Methodenaufrufe mit Parameterübergabe zur Manipulation von Objekteigenschaften (z.B. Farbe, Position, Größe) 		

Unterrichtsvorhaben EF-III

Thema: Grundlagen der Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand von grafischen Spielen mit Greenfoot und Simulationen mit LeJOS

Leitfragen: *Wie lassen sich Animationen und Simulationen optischer Gegenstandsbereiche unter Berücksichtigung von Tastatureingaben realisieren? Wie kann der Lego-Roboter mit seinen Aktoren (Motoren, Display) auf unterschiedliche Eingaben, durch Tasten, Sensoren reagieren? Wie lassen sich komplexe Datenflüsse und Beziehungen zwischen Objekten und Klassen realisieren?*

Vorhabenbezogene Konkretisierung:

Der Schwerpunkt dieses Unterrichtsvorhabens liegt auf der Entwicklung mehrerer Projekte, die durch Eingaben des Benutzers gesteuerte Animationen aufweisen.

Zunächst wird ein Projekt bearbeitet, in dem die Schülerinnen und Schüler die didaktische Entwicklungsumgebung Greenfoot kennenlernen.

Anhand dieses Projektes werden auch Zählschleifen und Verzweigungen eingeführt und an weiteren Beispielpunkten eingeübt. Dabei muss es sich nicht zwangsläufig um solche handeln, bei denen Kontrollstrukturen lediglich zur Animation verwendet werden. Auch die Erzeugung größerer Mengen grafischer Objekte und deren Verwaltung in einem Feld kann ein Anlass zur Verwendung von Kontrollstrukturen sein.

Ein erstes größeres Projekt ist das Spiel Schatzräuber. Dieses wird von den Schülern unter Anleitung Schritt für Schritt selber entwickelt und später selbstständig erweitert.

Das Unterrichtsvorhaben schließt mit einem Lego-Roboter-Projekt bei dem die Schüler zunächst über ein „Hello World“-Programm den Umgang mit LeJOS in BlueJ und das Übertragen von Programmen auf ein externes System kennenlernen. Anschließend wird mit einem Reaktionstest der Umgang mit Sensoren eingeübt, um zum Abschluss mithilfe von Schleifen und Verzweigungen den Lego-Roboter autonom durch ein Labyrinth fahren zu lassen.

Dieses Unterrichtsvorhaben beschäftigt sich im weiteren Verlauf im Schwerpunkt mit dem Aufbau komplexerer Objektbeziehungen. Während in vorangegangenen Unterrichtsvorhaben Objekte nur jeweils solchen Nachrichten schicken konnten, die sie selbst erstellt haben, soll in diesem Unterrichtsvorhaben diese hierarchische Struktur aufgebrochen werden.

Dazu bedarf es zunächst einer präzisen Unterscheidung zwischen Objektreferenzen und Objekten, so dass klar wird, dass Dienste eines Objektes von unterschiedlichen Objekten über unterschiedliche Referenzen in Anspruch genommen werden können. Auch der Aufbau solcher Objektbeziehungen muss thematisiert werden. Des Weiteren wird das Prinzip der Vererbung im objektorientierten Sinne angesprochen. Dazu werden die wichtigsten Varianten der Vererbung anhand von verschiedenen Projekten vorgestellt. Zunächst wird die Vererbung als Spezialisierung im Sinne einer einfachen Erweiterung einer Oberklasse vorgestellt. Darauf folgt ein Projekt, welches das Verständnis von Vererbung um den Aspekt der späten Bindung erweitert, indem Dienste einer Oberklasse überschrieben werden. Modellierungen sollen in Form von Implementationsdiagrammen erstellt werden.

Im Verlauf der Projekte kann kurz auf das Prinzip der abstrakten Klasse eingegangen werden. Dieser Inhalt ist aber nicht obligatorisch für die Einführungsphase.

Zeitbedarf: 18 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Grundlagen der Entwicklungsumgebung Greenfoot mithilfe von „Leafs and Wombats“</p> <ul style="list-style-type: none"> • Darstellung / Verhalten von Objekten (act-Methoden) • Grundlegende Methoden der Klassen Actor, Animal und World • Objektinteraktion 	<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), • ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekten oder lineare Datensammlungen zu (M), • ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M), • modifizieren einfache Algorithmen und Programme (I), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I), • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), • testen Programme schrittweise anhand von Beispielen (I), 	<ul style="list-style-type: none"> • Arbeitsblätter Greenfoot, BlueJ • Projekt „Leafs and Wombats“ • Projekt „Schatzräuber“ • LEGO und Lego-Roboter
<p>2. Implementierung und Erweiterung des Spiels „Schatzräuber“ unter Hinzunahme der</p> <ul style="list-style-type: none"> • Tastatursteuerung • Sounds • Bewegte Grafiken • Counter • Objekte erstellen neue Objekte 	<p>(siehe obiges Zeilenpaar)</p>	<p>(siehe obiges Zeilenpaar)</p>

<p>3. Implementierung von LeJOS-Programmen für Lego-Roboter</p> <ul style="list-style-type: none">• „Hello World“• Reaktionstest• Labyrinth-Rover	<ul style="list-style-type: none">• interpretieren Fehlermeldungen und korrigieren den Quellcode (!).	
---	---	--

Unterrichtsvorhaben EF-IV

Thema: Implementierung einfacher Algorithmen und GUI-Entwurf

Leitfrage: Welche Strategien aus dem Erfahrungsschatz der Schülerinnen und Schüler lassen sich algorithmisch beschreiben und programmieretechnisch realisieren? Wie lassen sich benutzerfreundliche Windowsanwendungen aus den Algorithmen entwickeln?

Vorhabenbezogene Konkretisierung:

Verschiedene Algorithmen (z. B. Freundschaftsalgorithmus (TBBT), ggT, kgV, Primfaktorzerlegung, Teilmenge, etc.) werden sprachlich formuliert, modifiziert und implementiert. Der Ablauf von Algorithmen wird z. B. mit Variablenbelegungstabellen dargestellt und somit die Korrektheit des Algorithmus verifiziert.

Mithilfe einer geeigneten Entwicklungsumgebung (z. B. JavaEditor, Netbeans, Eclipse, o. a.) werden grafische Oberflächen erstellt und mit den zuvor erstellten Algorithmen verknüpft. Dabei werden die Ereignisbehandlungsroutinen der Oberflächen-Objekte verwendet und das Geheimnisprinzip angewendet. Dabei findet das Model-View-Controller-Konzept Anwendung.

Zeitbedarf: 18 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Mathematische Algorithmen</p> <ul style="list-style-type: none"> Algorithmus zur ggT-Berechnung durch Schleifen mit Anfangs- und Endbedingung Wiederverwendbarkeit von Methoden bei der Berechnung des kgV komplexe Algorithmen zur Primfaktorzerlegung und Teilermengenberechnung 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekten [oder lineare Datensammlungen] zu (M), ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M), stellen den Zustand eines Objekts dar (D), analysieren und erläutern einfacher Algorithmen und Programme (A) 	<ul style="list-style-type: none"> Java-Editor o.ä. Arbeitsblätter
<p>2. Erstellung von Anwendungen mit grafischer Benutzeroberfläche</p> <ul style="list-style-type: none"> GUI-Entwurf Model-View-Controller-Konzept Darstellung von UML-Implementations-Diagrammen 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M), stellen die Kommunikation zwischen Objekten grafisch dar (M), stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D). <p>bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A).</p>	

Unterrichtsvorhaben EF-V

Thema: Entwurf und Implementierung einfacher Sortierverfahren und eines klassischen kryptologischen Verfahrens

Leitfragen: *Wie kann man die bisherigen Erkenntnisse im Umgang mit Arrays zum Sortieren von großen Datenmengen und zur einfachen „Verschlüsselung“ von Daten benutzen? Wie effizient ist ein Algorithmus hinsichtlich seines Laufzeitverhaltens?*

Vorhabenbezogene Konkretisierung:

Aufbauend auf den programmierertechnischen Grundlagen des letzten Unterrichtsvorhabens können die Schülerinnen und Schüler nun ihr Wissen zu Arrays und Strings kontextbezogen anwenden:

Dazu werden mindestens zwei der einfachen Sortierverfahren behandelt, indem sie simuliert, implementiert und getestet werden. In der anschließenden Analyse reflektieren die Schülerinnen und Schüler das Laufzeitverhalten der Algorithmen im best-, worst- und average-case.

Darüber hinaus wird ein einfaches kryptologisches Verfahren (z.B. CAESAR) behandelt. Neben der Durchführung der einfachen Verschlüsselungstechnik werden die zur Implementierung nötigen String-Operationen vertieft. Hierzu gehört auch die Thematisierung des ASCII-Codes. Die Schülerinnen und Schüler reflektieren die Sicherheitsaspekte des Verfahrens.

Zeitbedarf: 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Implementierung und Analyse einfacher Sortier- und Suchverfahren auf Arrays	<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 hinsichtlichlich Zeitaufwand und Speicherplatzbedarf (A). 	<p>Teams</p> <p>Apothekerwaagen & Streichholzschachteln</p> <p>MathePrisma Uni Wuppertal</p> <p>Youtube</p> <p>BlueJ</p>
2. Implementierung und Analyse eines einfachen kryptologischen Verfahrens	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern einfache Algorithmen und Programme (A), • implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen und Methodenaufrufen (!) • testen Programme schrittweise anhand von Beispielen (!). 	<p>Teams</p> <p>MathePrisma Uni Wuppertal</p> <p>Youtube</p> <p>BlueJ</p>

Unterrichtsvorhaben EF-VI

Thema: Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes

Leitfrage: *Welche Entwicklung durchlief die moderne Datenverarbeitung und welche Auswirkungen ergeben sich insbesondere hinsichtlich neuer Anforderungen an den Datenschutz daraus?*

Vorhabenbezogene Konkretisierung:

Das folgende Unterrichtsvorhaben stellt den Abschluss der Einführungsphase dar. Schülerinnen und Schüler sollen selbstständig informatische Themenbereiche aus dem Kontext der Geschichte der Datenverarbeitung und insbesondere den daraus sich ergebenden Fragen des Datenschutzes bearbeiten. Diese Themenbereiche werden in Kleingruppen bearbeitet und in Form von Plakatpräsentationen vorgestellt. Schülerinnen und Schüler sollen dabei mit Unterstützung des Lehrenden selbstständige Recherchen zu ihren Themen anstellen und auch eine sinnvolle Eingrenzung ihres Themas vornehmen.

Anschließend wird verstärkt auf den Aspekt des Datenschutzes eingegangen. Dazu wird das Bundesdatenschutzgesetz in Auszügen behandelt und auf schülernahe Beispielsituationen zur Anwendung gebracht. Dabei steht keine formale juristische Bewertung der Beispielsituationen im Vordergrund, die im Rahmen eines Informatikunterrichts auch nicht geleistet werden kann, sondern vielmehr eine persönliche Einschätzung von Fällen im Geiste des Datenschutzes.

Zeitbedarf: 15 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Selbstständige Erarbeitung von Themen durch die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> Mögliche Themen zur Erarbeitung in Kleingruppen: <ul style="list-style-type: none"> „Eine kleine Geschichte der Digitalisierung: vom Morsten zum modernen Digitalcomputer“ „Eine kleine Geschichte der Kryptographie: von Caesar zur Enigma“ „Von Nullen, Einsen und mehr: Stellenwertsysteme und wie man mit ihnen rechnet“ „Kodieren von Texten und Bildern: ASCII, RGB und mehr“ „Auswirkungen der Digitalisierung: Veränderungen der Arbeitswelt und Datenschutz“ <p>Vorstellung und Diskussion durch Schülerinnen und Schüler</p>	<ul style="list-style-type: none"> Die Schülerinnen und Schüler <ul style="list-style-type: none"> bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informationssystemen (A), erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A), stellen ganze Zahlen und Zeichen in Binärcodes dar (D), interpretieren Binärcodes als Zahlen und Zeichen (D), nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation. (K). 	<p>Beispiele, Medien, Materialien</p> <p><i>Beispiel:</i> Ausstellung zu informatischen Themen</p> <p>Die Schülerinnen und Schüler bereiten eine Ausstellung zu informatischen Themen vor. Dazu werden Stellwände und Plakate vorbereitet, die ggf. auch außerhalb des Informatikunterrichts in der Schule ausgestellt werden können.</p> <p><i>Materialien:</i></p> <p>Schülerinnen und Schüler recherchieren selbstständig im Internet, in der Schulbibliothek, in öffentlichen Bibliotheken, usw.</p>
<p>2. Vertiefung des Themas Datenschutz</p> <ul style="list-style-type: none"> Erarbeitung grundlegender Begriffe des Datenschutzes Problematisierung und Anknüpfung an die Lebenswelt der Schülerinnen und Schüler Diskussion und Bewertung von Fallbeispielen aus dem Themenbereich „Datenschutz“ 		<p><i>Beispiel:</i> Fallbeispiele aus dem aktuellen Tagesgeschehen</p> <p>Die Schülerinnen und Schüler bearbeiten Fallbeispiele aus ihrer eigenen Erfahrungswelt oder der aktuellen Medienberichterstattung.</p> <p><i>Materialien:</i></p> <p>Materialblatt zum Bundesdatenschutzgesetz (Download EF-VI.1)</p>

II) / III) Qualifikationsphase – GRUNDKURS/LEISTUNGSKURS

Die Aufstellung der Unterrichtsvorhaben erfolgt für den Grundkurs und den **Leistungskurs** kombiniert.

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 bereitgestellte Informatiksystem und das Internet reflektiert zur Erschließung, Aufbereitung und Präsentation fachlicher Inhalte (D),
- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung von Dateien unter Berücksichtigung der Rechteverwaltung / **Daten, zur Organisation von Arbeitsabläufen sowie zur Verteilung und Zusammenführung von Arbeitsanteilen** (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).

Ebenso bieten fast alle Unterrichtsvorhaben, in denen Programme implementiert werden, die Gelegenheit, die folgenden Kompetenzen zu erwerben bzw. zu vertiefen:

Die Schülerinnen und Schüler

- nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I)
- beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),
- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),
- wenden eine didaktisch orientierte Entwicklungsumgebung (**didaktisch orientierte Entwicklungsumgebungen**) zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I),

- entwickeln mit didaktisch orientierten Entwicklungsumgebungen einfache Benutzungsoberflächen zur Kommunikation mit einem Informationssystem (M).

Unterrichtsvorhaben Q1-I:

Thema: Rekursive Algorithmen und Backtracking in Anwendungskontexten

Leitfragen: *Wie können komplexe, rekursiv definierte Probleme informatisch gelöst werden? Gibt es schnelle (rekursiv definierte) Sortier- und Suchverfahren?*

Vorhabenbezogenen Konkretisierung:

Ausgehend von einem Problem wie z. B. "Türme von Hanoi" wird Rekursion als fundamentale Idee der Informatik zunächst im mathematischen, danach aber auch im informatischen Zusammenhang angewendet. Dabei wird zwischen linearen und verzweigten Rekursionen unterschieden und das Laufzeitverhalten bei hoher Rekursionstiefe analysiert.

Verschiedene NP-vollständige Probleme (wie z. B. Rucksack, n-Damen, Springer, Irrgarten, etc.) werden algorithmisch rekursiv formuliert und als Backtracking-Algorithmus implementiert.

Bereits bekannte Such- und Sortierverfahren (z. B. Sortieren durch Einfügen, Sortieren durch Auswahl, Sequentielle Suche) werden rekursiv formuliert und durch leistungsfähigere Verfahren (z. B. Quicksort, Mergesort, Heapsort, Binäre Suche) ergänzt. Die neuen Verfahren werden implementiert.

Zeitbedarf: 20/30 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Entwicklung der Rekursion als fundamentale Idee der Informatik <ul style="list-style-type: none"> • Rekursive Formeln • Rekursive Funktionen/Methoden • Rekursive Programmierung 	Die Schülerinnen und Schüler <ul style="list-style-type: none"> • Analysieren und erläutern Algorithmen und Programme (A), • Modifizieren Algorithmen und Programme (I) 	Türme von Hanoi mit Schwerpunkt auf <ul style="list-style-type: none"> • Zahl der Versetzungsoperationen • Protokollierung der Versetzungen

<p>2. Rekursion in mathematischen und informati- schen Kontexten</p> <ul style="list-style-type: none"> • Rekursion in mathematische Kontexten • Analyse und Darstellung des rekursiven Ablaufs einer Methode • Analyse des Laufzeitverhaltens linearer und verzweigter Rekursion 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • Analysieren und erläutern Algorithmen und Programme (A), • Modifizieren Algorithmen und Programme (I), • Stellen iterative und rekursive Algorithmen umgangssprachlich dar (D), • Testen Programme systematisch anhand von Beispielen und mit Hilfe von Testanwendungen (I), • Untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A). 	<ul style="list-style-type: none"> • Fakultätsfunktion (lineare Rekursion) • Fibonacci-Funktion (verzweigte Rekursion) • ggT (verzweigte Rekursion) • einfache Rekursive Grafiken (Farn etc.) • ggf. Fraktale (Kochkurve, Sierpinski dreieck, etc.)
<p>3. NP-vollständige Probleme lösen mit Backtracking</p> <ul style="list-style-type: none"> • Erarbeitung verschiedener NP-vollständiger Probleme • Algorithmische Beschreibung einer Lösungs- idee • Implementierung eines Problems als Backtrackingalgorithmus 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern Algorithmen und Programme (A), • modifizieren Algorithmen und Programme (I), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ und "Backtracking" (M) • testen Programme systematisch anhand von Beispielen und mit Hilfe von Testanwendungen (I). • untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A). 	<ul style="list-style-type: none"> • Demonstrationsprogramm zur Visualisierung von Sortierverfahren • Quicksortvisualisierung zur Erarbeitung der Idee • Suchspiel zur Erarbeitung der Binären Suche

Unterrichtsvorhaben Q1-II:

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

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

Vorhabenbezogene Konkretisierung:

Nach Analyse einer Problemstellung in einem geeigneten Anwendungskontext, in dem Daten nach dem Last-In-First-Out-Prinzip verwaltet werden, werden der Aufbau von Stapeln am Beispiel dargestellt und die Operationen der Klasse Stack anhand der Abiturklasse erläutert. Anschließend werden für die Anwendung notwendige Klassen modelliert und implementiert.

Anschließend wird die Anwendung modifiziert, um den Umgang mit der Datenstruktur zu üben. Anhand einer Anwendung, in der Daten nach dem First-In-First-Out-Prinzip verwaltet werden, werden Unterschiede zwischen den Datenstrukturen Schlange und Stapel erarbeitet. Dabei werden die Operationen der Datenstruktur Schlange thematisiert und die entsprechende Abiturklasse Queue verwendet.

Um einfacher an Objekte zu gelangen, die zwischen anderen gespeichert sind, wird die Klasse List gemäß der Abiturklasse eingeführt und in einem Anwendungskontext verwendet.

(Je nach thematisierten Anwendungskontexten ist eine andere Reihenfolge bei der Behandlung der oben genannten Datenstrukturen möglich und zulässig.)

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: 25 / 30 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse Stack</p> <ul style="list-style-type: none">Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und OperationenErarbeitung der Funktionalität der Klasse StackModellierung und Implementierung der Anwendung unter Verwendung der Klasse Stack.	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D),	Visualisierungsprogramm zu dynamischen Datenstrukturen

<ul style="list-style-type: none"> ● Implementierung der Klasse Stack 	<ul style="list-style-type: none"> ● modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), ● ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekten sowie lineare und nichtlineare Datensammlungen zu (M), ● ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), ● stellen die Kommunikation zwischen Objekten grafisch dar (D), ● stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), ● dokumentieren Klassen (D), ● analysieren und erläutern objektorientierte Modellierungen (A), ● implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I). ● analysieren und erläutern Algorithmen und Programme (A), ● modifizieren Algorithmen und Programme (I), ● stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D), ● implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), 	<p>Visualisierungsprogramm zu dynamischen Datenstrukturen</p>
<p>2. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse Queue</p> <ul style="list-style-type: none"> ● Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen ● Erarbeitung der Funktionalität der Klasse Queue ● Modellierung und Implementierung der Anwendung unter Verwendung der Klasse Queue. ● Implementierung der Klasse Queue 		
<p>3. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse List</p> <ul style="list-style-type: none"> ● Umwandlung der Queue in eine PriorityQueue, welche auf der Datenstruktur List basiert ● Erarbeitung der Vorteile der Klasse List im Gegensatz zu den bereits bekannten linearen Strukturen ● Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse List. ● Implementierung der Klasse List 		<p>Visualisierungsprogramm zu dynamischen Datenstrukturen Einfache Oberfläche zur Visualisierung der linearen Liste, z.B. Verwaltung Ergebnisliste</p>

	<ul style="list-style-type: none"> • testen Programme systematisch anhand von Beispielen und mit Hilfe von Testanwendungen (I); • erläutern Operationen dynamischer (linearer oder/und nicht-linearer) Datenstrukturen (A), • implementieren Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (I), 	
<p>4. Vertiefung / Anwendung einer linearen Datenstruktur im Anwendungskontext.</p>	<p>zusätzlich: Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie (M), 	<p>Umsetzung in einem größeren Projekt</p>

Unterrichtsvorhaben Q1-III:

Thema: Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen am Beispiel der Binärbäume

Leitfrage: *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 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 Baum Inhalte 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 Vorgaben für das Zentralabitur weitere Klassen oder Methoden in diesem Kontext modelliert und implementiert. Die Suchbäume werden wie zuvor auch grafisch dargestellt.

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

Zeitbedarf: 20/25 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Analyse von Baumstrukturen in verschiedenen Kontexten	Die Schülerinnen und Schüler	<ul style="list-style-type: none">Arbeitsmaterial zum Biber Suchbaum

<ul style="list-style-type: none"> • Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit) • Aufbau und Darstellung von binären Bäumen anhand von Baumstrukturen in verschiedenen Kontexten 	<ul style="list-style-type: none"> • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D), • modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekten sowie/oder lineare und nichtlineare Datensammlungen zu (M), • verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie (M), • ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), • stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen (D), • analysieren und erläutern objektorientierte Modellierungen (A), 	<ul style="list-style-type: none"> • Arbeitsmaterial zum Biber Suchbaum
<p>2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse BinaryTree</p> <ul style="list-style-type: none"> • Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext • Modellierung eines Entwurfdiagramms und Entwicklung eines Implementationsdiagramms • Erarbeitung der Klasse BinaryTree und beispielhafte Anwendung der Operationen • Implementierung der Anwendung oder von Teilen der Anwendung • Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf • Implementierung der Standardoperationen der Klasse BinaryTree 		<ul style="list-style-type: none"> • Arbeitsmaterial BinärerSuchbaum • Arbeitsmaterial Adressbuch
<p>3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse BinarySearchTree</p> <ul style="list-style-type: none"> • Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen • Modellierung eines Entwurfdiagramms und Entwicklung eines Implementationsdiagramms, Grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften • Erarbeitung der Klasse BinarySearchTree und Einführung des Interface Item zur Realisierung einer geeigneten Ordnungsrelation 		

<ul style="list-style-type: none"> • Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums • Implementierung ausgewählter Standardoperationen der Klasse BinarySearchTree 	<ul style="list-style-type: none"> • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I). • analysieren und erläutern Algorithmen und Programme (A), • modifizieren Algorithmen und Programme (I), • stellen iterative und rekursive Algorithmen umfangsprachlich und grafisch dar (D), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ und „Backtracking“ (M), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • testen Programme systematisch anhand von Beispielen und mit Hilfe von Testanwendungen (I). • erläutern Operationen dynamischer (linearer oder/und nicht-linearer) Datenstrukturen (A), • implementieren Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (I), 	<p>Arbeitsmaterial Huffman-Codes</p>
<p>4. Übung und Vertiefung der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen</p>		

Unterrichtsvorhaben Q1-IV:

Thema: Sicherheit und Datenschutz in Netzstrukturen und Modellierung und Implementierung von Client-Server-Anwendungen

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

Vorhabenbezogene Konkretisierung:

Ausgehend von einer Kommunikation zwischen zwei Kommunikationspartnern über eine einfache Leitung wird die Notwendigkeiten einer Datenübertragung erarbeitet. Die Schichten des TCP/IP-Schichtenmodells werden beispielgebunden erarbeitet (Basisbandübertragungsverfahren, Prüfverfahren, Vermittlungsschicht, Anwendungsprotokoll) und an einer Simulationssoftware getestet. Verschiedene Netzwerk-Topologien werden entwickelt und in Client-Server-Anwendungen simuliert.

Im Leistungskurs werden ausgehend von einer einfachen Echo-Anwendung die beteiligten Komponenten (EchoSERVER und Echoclient) entwickelt und unter Verwendung der ZA-Klassen implementiert.

Die Echo-Anwendung wird zu einer Chat-Anwendung erweitert, notwendige Protokolle werden entwickelt und systematisch dargestellt.

Die Schülerinnen und Schüler entwickeln eine individuelle Client-Serveranwendung, definieren notwendige Protokolle und erweitern die Chat-Anwendung entsprechend der Vorgaben.

Über die Sicherheit von Netzwerk Anwendungen wird das Augenmerk auf verschiedene symmetrische und asymmetrische kryptografische Verfahren gelenkt, welche analysiert und erläutert werden. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht runden das Unterrichtsvorhaben ab.

Zeitbedarf: 15 / 25 Stunden

Lernmittel /Materialien:

- Arbeitsblätter zur Einführung in Netzwerke
- Simulationsprogramm "Filius"
- Arbeitsblätter und Skript zu "Filius"
- Arbeitsblätter zum Chat-Projekt
- Kryptografie-Programm "Cryptool"
- Arbeitsblätter zu kryptografischen Verfahren

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Schichten des TCP/IP-Protokolls</p> <ul style="list-style-type: none"> • Erarbeitung der Notwendigkeiten einer Netzwerkkommunikation • Erarbeitung der Schichten des TCP/IP-Protokolls: Ethernet-, Internet, Transport- und Anwendungsschicht 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • Beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A), 	<ul style="list-style-type: none"> • Arbeitsblätter zur Einführung in Netzwerke
<p>2. Simulation von Netzwerken / Netzwerk-Topologien</p> <ul style="list-style-type: none"> • Erarbeitung der Topologien: Peer-to-Peer, Sterntopologie, Baumtopologie, Vermaschtes Netz • Simulation von Client-Server-Anwendungen • Simulation von Protokollen der Anwendungsschicht (POP3, SMTP, etc.) 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • Beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A), • Analysieren und erläutern Protokolle zur Kommunikation in einem Client-Server-Netzwerk (A), 	<ul style="list-style-type: none"> • Simulationssoftware FILLIUS • Arbeitsblätter und Skript zu FILLIUS
<p>3. Entwicklung verschiedener Client-Server-Anwendungen</p> <ul style="list-style-type: none"> • Grundlegende Begriffe • ZA-Klassen • Echo-Anwendung • Chat Anwendung (Nutzung der List) • Beliebige Client-Server-Anwendung 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • erläutern das Prinzip der Nebenläufigkeit (A), • analysieren und erläutern Algorithmen und Methoden zur Client-Server-Kommunikation (A), • entwickeln und implementieren Algorithmen und Methoden zur Client-Server-Kommunikation (I). • analysieren und erläutern Protokolle zur Kommunikation in einem Client-Server-Netzwerk (A), 	<p>Umsetzung eines größeren Projektes</p> <ul style="list-style-type: none"> • Chat • Anwendungsserver (wie z.B. Buchungsserver für Kinokarten)

<p>4. Analyse und Erläuterung kryptografischer Verfahren</p> <ul style="list-style-type: none"> • Erläuterung symmetrischer Verfahren: monoalphabetisch: Cäsar, polyalphabetisch: Vigenère • Erläuterung asymmetrischer Verfahren: RSA, Diffie-Hellman • Analyse der Sicherheit verschiedener Verfahren und Auswirkungen auf den Datenschutz/Urheberrecht 	<ul style="list-style-type: none"> • entwickeln und erweitern Protokolle zur Kommunikation in einem Client-Server-Netzwerk (M). <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern Algorithmen und Programme (A), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D), • analysieren und erläutern Eigenschaften, Funktionsweisen und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A). • untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A), • untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A). 	<ul style="list-style-type: none"> • Arbeitsblätter zu kryptografischen Verfahren (SpiroCamp Uni Wuppertal) • CryptTool • Materialien von Klicksafe (Zusatzmodule „Nicht alles was geht, ist auch erlaubt“, „Ich bin öffentlich ganz privat“, „Datenschutz im Internet“)
--	--	---

Unterrichtsvorhaben Q2-I:

Hinweis: Im Leistungskurs sollte dieses Thema in die Q1 vorgezogen werden.

Thema: Modellierung und Nutzung relationaler 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 konkreten Anwendungssituation entwickeln die Schülerinnen und Schüler Ideen zur Modellierung von Daten und erkennen die Vorzüge von Datenbanksystemen.

In weiteren Anwendungskontexten müssen Datenbanken 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 das Relationale Modell überführt.

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.

Ausgehend von einer vorhandenen Datenbasis, für die eine Datenbank angelegt und mit den Daten gefüllt wird, 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.

Mit Hilfe von SQL-Anweisungen können anschließend im Kontext relevante Informationen aus der Datenbank extrahiert werden. Die Operationen der Relationenalgebra werden mit SQL-Abfragen simuliert.

Anhand von Fallbeispielen werden Probleme bei der Nutzung von Datenbanksystemen aufgezeigt und im Hinblick auf gesellschaftliche Auswirkungen diskutiert.

(Die exakte Abfolge der Teilvorhaben sowie die Verzahnung theoretischer und praktischer Anteile kann hierbei je nach Anwendungskontext flexibel angepasst werden.)

Lernmittel / Materialien:

40

- Arbeitsblätter und Demonstrationsprogramme in Teams
- Datenbankserver (z. B. XAMPP)
- SQL-Tutorial (z. B. die Seite VideoCenter-Datenbank)
- Schönigh Informatik II (Online-Buchhandel)

Zeitbedarf: 25 / 30 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Nutzung von relationalen Datenbanken</p> <ul style="list-style-type: none"> • Entity-Relationship-Diagramm: Ermittlung von Entitäten, zugehörigen Attributen, Relationen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms Er-läuterung und Modifizierung einer Datenbankmodellierung • Aufbau von Datenbanken und Grundbegriffe: Entwicklung von Fragestellungen zur vorhandenen Datenbank Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Be-griffe Tabelle, Attribut, Datensatz, Datentyp, Pri-märschlüssel, Fremdschlüssel, Datenbank-schema • SQL-Abfragen: Analyse vorgegebener SQL-Abfra-gen und Erarbeitung der Sprachelemente von 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M), • stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten mit Kardinalitäten in einem Entity-Relationship-Diagramm grafisch dar (D), • bestimmen Primär- und Sekundärschlüssel (M), • implementieren ein relationales Datenbank-schema als Datenbank (I), • analysieren und erläutern eine Datenbankmodellierung (A), • ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D), • analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A), 	<p>VideoCenter Datenbank</p>

<p>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>	<ul style="list-style-type: none"> • verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I). • erläutern Eigenschaften und Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A), 	
<p>2. Modellierung von relationalen Datenbanken</p> <ul style="list-style-type: none"> • Entwicklung einer Datenbank aus einem Datenbankentwurf: Modellierung eines relationalen Datenbankschemas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln • 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) • Vertiefung an einem weiteren Datenbankbeispiel: Vertiefungen am Beispiel der Relationalgebra 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • modifizieren eine Datenbankmodellierung (M), • modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M), • bestimmen Primär- und Sekundärschlüssel (M), • analysieren und erläutern eine Datenbankmodellierung (A), • erläutern die Eigenschaften normalisierter Datenbankschemata (A), • überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D), •überführen Datenbankschemata in die 1. bis 3. Normalform (M). 	<p>Schöningh Informatik II (Online-Buchhandel)</p> <p>ZA-Klassen</p>
<p>3. Gesellschaftliche Auswirkungen der Nutzung von Datenbanksystemen</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informationssystemen sowie Aspekte der Sicherheit 	<ul style="list-style-type: none"> • Fallbeispiele zur Nutzung von Datenbanksystemen

	<p>von Informationssystemen, des Datenschutzes und des Urheberrechts (A),</p> <ul style="list-style-type: none"> • untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informationssystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A). 	<ul style="list-style-type: none"> • Spiel zum Missbrauch von Daten: DataDealer (http://datadealer.com/de) • Forbildungsmaterial Yoda-Datenbank
--	---	--

Unterrichtsvorhaben Q2-II:

Thema: Endliche Automaten und formale Sprachen

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

Vorhabenbezogene Konkretisierung:

Ausgehend von einem konkreten Anwendungsbeispiel entwickeln die Schülerinnen und Schüler das Modell der Grammatik einer formalen Sprache und das Modell des endlichen Automaten. Die Schülerinnen und Schüler überführen Automaten in verschiedene Darstellungsformen und ermitteln die akzeptierte Sprache eines Automaten (z. B. in Form von regulären Ausdrücken). An einem Beispiel wird ein nichtdeterministischer Akzeptor als Alternative gegenüber einem entsprechenden deterministischen Akzeptor eingeführt.

Der Zusammenhang zwischen endlichen Automaten und regulären Grammatiken wird durch die Entwicklung allgemeingültiger Verfahren zur Transformation zwischen Automat und Grammatik dargestellt. Die Unzulänglichkeit endlicher Automaten und regulärer Grammatiken wird an Beispielen verdeutlicht.

Das Grammatikmodell der regulären Grammatiken wird auf das Modell der kontextfreien Grammatiken erweitert und die Auswirkungen auf das entsprechende Automatenmodell der Kellerautomaten veranschaulicht. Die Unzulänglichkeit der Kellerautomaten und kontextfreien Grammatiken wird an Beispielen verdeutlicht.

Ausgehend von einer einfachen formalen Sprache (z. B. reduziertes Java) werden die Bestandteile eines Compilers dargestellt:

Der Scanner eines Compilers wird in Form eines endlichen Automaten modelliert und implementiert. Die Begriffe der Symboltabelle und der Tokenliste werden inhaltlich gefüllt. Der Sprachumfang der einfachen formalen Sprache wird leicht erweitert und die Auswirkungen auf den Automaten und die Implementierung wird beobachtet.

Der Parser eines Compilers wird in Form einer kontextfreien Grammatik modelliert und implementiert. Der Sprachumfang der einfachen formalen Sprache wird um weitere Regeln ergänzt und der Parser wird angepasst.

Bei Bedarf wird ein Übersetzer-Modul entwickelt, welches die einfache formale Sprache in eine Sprachebene übersetzt (z. B. interpretiert und compiliert).

Zeitbedarf: 25/40 Stunden

Lernmittel / Materialien:

- Arbeitsblätter und Demonstrationsprogramme in der Teams-Plattform
- Pirates Island
- Automaten simulationsprogramm (z. B. JFlap, Exorciser)

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p>1. Einführung in Automaten/Grammatiken</p> <ul style="list-style-type: none"> • Grammatiken: Grammatik einer natürlichen Sprache • Grammatik einer künstlichen Sprache • Idee des Parsens • Automaten: erkennender Automat zu Symbolen einer Sprache • Modell des endlichen Automaten • Darstellungsformen Sprache eines Automaten • als regulärer Ausdruck nichtdeterministische Automaten 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern die Eigenschaften endlicher Automaten und Kellerautomaten einschließlich ihres Verhaltens bei bestimmten Eingaben (A), • ermitteln die Sprache, die ein endlicher Automat oder ein Kellerautomat akzeptiert (D), • entwickeln und modifizieren zu einer Problemstellung endliche Automaten oder Kellerautomaten (M), 	<p>Einstieg Pirates Island</p> <p>Weitere Arbeitsblätter</p>

	<ul style="list-style-type: none"> stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D), entwickeln zur Grammatik einer regulären oder kontextfreien Sprache einen zugehörigen endlichen Automaten oder einen Kellerautomaten (M). 	
<p>2. Zusammenhang zwischen endlichen Automaten und regulären Grammatiken</p> <ul style="list-style-type: none"> reguläre Grammatik: Definition Anwendungen Zusammenhang zu endlichen Automaten Grenzen der endlichen Automaten/regulären Grammatiken 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> analysieren und erläutern Grammatiken regulärer und kontextfreier Sprachen (A), modifizieren Grammatiken regulärer und kontextfreier Sprachen (M), ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A), entwickeln zu einer regulären oder kontextfreien Sprache eine Grammatik, die die Sprache erzeugt (M), entwickeln zur akzeptierten Sprache eines Automaten eine zugehörige Grammatik (M), beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D), zeigen/erläutern die Grenzen endlicher Automaten und regulärer Grammatiken/Sprachen im Anwendungszusammenhang auf (A). 	Arbeitsblätter
<p>3. Zusammenhang zwischen Kellerautomaten und kontextfreien Grammatiken</p> <ul style="list-style-type: none"> kontextfreie Grammatik: Definition Anwendungen 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> analysieren und erläutern die Eigenschaften endlicher Automaten und Kellerautomaten 	

<ul style="list-style-type: none"> • Modell des Kellerautomaten Definition Darstellungsfomen Anwendungen / Sprache eines Kellerautomaten • Zusammenhang zwischen Kellerautomaten/kontextfreien Grammatiken • Grenzen der Kellerautomaten 	<p>einschließlich ihres Verhaltens bei bestimmten Eingaben (A),</p> <ul style="list-style-type: none"> • ermitteln die Sprache, die ein endlicher Automat oder ein Kellerautomat akzeptiert (D), • entwickeln und modifizieren zu einer Problemstellung endliche Automaten oder Kellerautomaten (M), • entwickeln zur Grammatik einer regulären oder kontextfreien Sprache einen zugehörigen endlichen Automaten oder einen Kellerautomaten (M), • analysieren und erläutern Grammatiken regulärer und kontextfreier Sprachen (A), • modifizieren Grammatiken regulärer und kontextfreier Sprachen (M), • entwickeln zu einer regulären oder kontextfreien Sprache eine Grammatik, die die Sprache erzeugt (M), • beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D). 	
<p>4. Die Schritte eines Compilers</p> <ul style="list-style-type: none"> • Scanner: endlicher Automat als Grundlage Symboltabelle und Tokenliste zur Verwaltung und Erkennung des Quelltextes Erweiterung des terminalen Alphabets der zu compilierenden formalen Sprache Implementierung als endlicher Automat 	<ul style="list-style-type: none"> • modellieren und implementieren Scanner, Parser und Interpreter zu einer gegebenen regulären Sprache (I). 	<p>Arbeitsblätter und Skript zum Compilerbau über Teams</p>

<ul style="list-style-type: none"> Parser: kontextfreie Grammatik als Grundlage Erweiterung des Sprachumfangs Implementierung als kontextfreie Grammatik 		
---	--	--

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: 10 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
1. Von-Neumann-Architektur und die Ausführung maschinennaher Programme <ul style="list-style-type: none"> prinzipieller Aufbau einer von Neumann-Architektur mit CPU, Rechenwerk, Steuerwerk, Register und Hauptspeicher 	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), 	Beispiel: Addition von 4 zu einer eingegeben Zahl mit einem Rechnermodell Materialien:

<ul style="list-style-type: none"> • einige maschinennahe Befehle und ihre Repräsentation in einem Binär-Code, der in einem Register gespeichert werden kann • Analyse und Erläuterung der Funktionsweise eines einfachen maschinennahen Programms 	<ul style="list-style-type: none"> • untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A). 	<p>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 –Von-Neumann-Architektur und maschinennahe Programmierung (Download Q2-III.1)</p>
<p>2. Grenzen der Automatisierbarkeit</p> <ul style="list-style-type: none"> • Vorstellung des Halteproblems • Unlösbarkeit des Halteproblems • Beurteilung des Einsatzes von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen 		<p><i>Beispiel:</i> Halteproblem</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 - Halteproblem (Download Q2-III.2)</p>

Unterrichtsvorhaben Q2-IV:

Thema: Modellierung und Implementierung dynamischer nichtlinearer Datenstrukturen am Beispiel der Graphen

Leitfragen: *Bei welchen Problemstellungen reichen die bekannten Datenstrukturen nicht aus? Welche Möglichkeiten gibt es, flexibel miteinander verknüpfte Informationen zu verwalten? Wie hängen die Datenstrukturen Graph, Baum und Liste zusammen?*

Vorhabenbezogene Konkretisierung:

Nach Analyse einer Problemstellung in einem geeigneten Anwendungskontext (z. B. das Eulerkreisproblem), in dem Daten in Form eines Graphen verwaltet werden, werden der Aufbau und die Darstellungsformen von Graphen am Beispiel dargestellt und ausgewählte Problemstellungen exemplarisch analysiert.

Die Operationen der Klasse Graph werden erläutert und im Anwendungszusammenhang bei der Lösung grundlegender Probleme (wie z.B. der Traversierung eines Graphen) genutzt.

Zeitbedarf: 15 Stunden

Lernmittel / Materialien:

- Arbeitsblätter in der moodle-Plattform
- eine didaktische Entwicklungsumgebung (z. B. Java-Editor o. a.)

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p>1. Die Datenstruktur des Graphen im Anwendungskontext</p> <ul style="list-style-type: none">• Grundbegriffe (Knoten, Kanten und anderes)• Darstellungsformen eines Graphen	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">• ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D),	<p>Eulerkreisproblem:</p> <ul style="list-style-type: none">• Arbeitsblätter zum Eulerweg/Eulerkreis• Visualisierungsprogramm für Graphen

	<ul style="list-style-type: none"> • modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), • verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie (M), • ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), • stellen die Kommunikation zwischen Objekten grafisch dar (D), • stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen (D), • analysieren und erläutern objektorientierte Modellierungen (A), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I). • analysieren und erläutern Algorithmen und Programme (A), • modifizieren Algorithmen und Programme (I), 	
--	---	--

	<ul style="list-style-type: none"> • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ und „Backtracking“ (M), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • testen Programme systematisch anhand von Beispielen und mit Hilfe von Testanwendungen (I). • erläutern Operationen dynamischer (linearer oder/und nicht-linearer) Datenstrukturen (A), 	
<p>2. Die Datenstruktur des Graphen im Anwendungskontext unter Nutzung der Klasse Graph</p> <ul style="list-style-type: none"> • Modellierung von Anwendungssituationen als Graph • Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen • Erarbeitung der Funktionalität der Klasse Graph • Modellierung und Implementierung verschiedener Problemstellungen unter Verwendung der Klasse Graph. 		<p>Projektarbeit im größeren Kontext</p> <ul style="list-style-type: none"> • Navigationssystem • Weiterführung des Eulerkreisproblems <p>Arbeitsblätter Teams</p> <ul style="list-style-type: none"> • Tiefensuche / Breitensuche • kürzeste Wege mit Dijkstra <p>mögliche Vertiefungen:</p> <ul style="list-style-type: none"> • Traveling Salesman Problem • minimale Spannäume

2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

In Absprache mit der Lehrerkonferenz sowie unter Berücksichtigung des Schulprogramms hat die Fachkonferenz Informatik des Nicolaus-Cusanus-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:

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

Fachliche Grundsätze:

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

2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

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

2.3.1 Beurteilungsbereich Klausuren

Verbindliche Absprachen:

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

Instrumente:

- Einführungsphase: 1 Klausur je Halbjahr
Dauer der Klausur: 90 Minuten
- Grund- und Leistungskurse Q 1: 2 Klausuren je Halbjahr
Dauer der Klausuren: 90 Minuten (Grundkurs), 135 Minuten (Leistungskurs)
- Grund- und Leistungskurse Q 2.1: 2 Klausuren
Dauer der Klausuren: 135 Minuten (Grundkurs), 180 Minuten (Leistungskurs)
- Grund- und Leistungskurse Q 2.2: 1 Klausur unter Abiturbedingungen
- Anstelle einer Klausur kann gemäß dem Beschluss der Lehrerkonferenz in Q 1.2 eine Facharbeit geschrieben werden.

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

Kriterien

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

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

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

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

2.3.2 Beurteilungsbereich Sonstige Mitarbeit

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

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 oder ein entsprechendes Projekt, 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 Leistungskursfach in der Qualifikationsphase.

2.4 Lehr- und Lernmittel

Eingesetzte Lehrbücher und Arbeitsmaterialien:

- Skripte und Arbeitsblätter
- Informatik, Lehrwerk für die gymnasiale Oberstufe – Neubearbeitung, Schöningh-Verlag
- Arbeitsblätter und Programmvorlagen auf der schuleigenen Lernplattform *Teams*

Eingesetzte Software (jeweils in der aktuellen Version):

- Java SDK
- BlueJ
- Greenfoot
- Java-Editor
- JFlap
- Excorsiser
- XAMPP
- Cryptool
- Weitere Demonstrationsprogramme

3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen

Die Fachschaft Informatik nutzt die sich flexibel ergebenden Gelegenheiten, mit anderen Fächern zu kooperieren.

Falls es sich anbietet, werden Exkursionen z.B. in die Fachhochschule der Wirtschaft (FHDW Bergisch Gladbach) durchgeführt.

4 Qualitätssicherung und Evaluation

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

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

Jeweils vor Beginn eines neuen Schuljahres werden in einer Sitzung der Fachkonferenz für die nachfolgenden Jahrgänge zwingend erforderlich erscheinende Veränderungen diskutiert und ggf. beschlossen, um erkannten ungünstigen Entscheidungen schnellstmöglich entgegenwirken zu können.