Schulinterner Lehrplan
der Holzkamp-Gesamtschule Witten zum
Kernlehrplan für die gymnasiale Oberstufe

Informatik

Abiturjahrgang 2025

Inhalt

1 Die Fachgruppe Informatik der Holzklamp-Gesamtschule Witten

Bei der Holzkamp-Gesamtschule handelt es sich um eine fünf- bis sechszügige Schule in Witten-Annen mit zurzeit ca. 1100 Schülerinnen und Schülern und ca. 100 Lehrerinnen und Lehrern. Das Einzugsgebiet der Schule umfasst das Stadtgebiet Witten und Herdecke, sowie Orte im näheren Umfeld.
Das Fach Informatik wird an der Holzkamp-Gesamtschule in der Jahrgangsstufe 6 als ITG (Informationstechnische Grundbildung) und ab der Jahrgangsstufe 8 im Ergänzungsstundenbereich (ES) zweistündig unterrichtet und jährlich von etwa 20-40 Schülerinnen und Schüler besucht, je nachdem ob zusätzlich ein Kurs „Bürotechnik“ angewählt wird. In der jeweils halbjährig unterrichteten Laufzeit dieser Kurse wird in altersstufengerechter Weise unter anderem auf Grundlagen der Algorithmik am Beispiel einer didaktischen Lernumgebung, auf Grundlagen der Datenverarbei- tung, technische Informatik und Grundlagen der Datenbankentwicklung eingegan- gen. Der Unterricht in der Sekundarstufe I erfolgt aufgrund der besonderen Situation in den Ergänzungsstunden halbjährlich organisiert, sodass die einzelnen Themen nicht aufeinander aufbauen. Dieser Unterricht erfolgt häufig in Form von größeren Projekten, die von den SuS durchgeführt werden. Hierbei wird versucht, auch Inhalte der Fächer Mathematik, Physik und Deutsch aufzugreifen und für die Projekte zu nutzen.

In der Sekundarstufe II bietet die Holzkamp-Gesamtschule für die eigenen Schülerin- nen und Schüler in allen Jahrgangsstufen jeweils einen Grundkurs in Informatik an.

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. In der Einführungsphase kommt dabei zusätzlich eine didaktische Bibliothek zum Einsatz, welche das Erstellen von grafischen Programmen erleichtert.

Durch projektartiges Vorgehen, offene Aufgaben und Möglichkeiten, Problemlösun- gen zu verfeinern oder zu optimieren, entspricht der Informatikunterricht der Oberstu- fe in besonderem Maße den Erziehungszielen, Leistungsbereitschaft zu fördern, oh- ne zu überfordern.

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

Zurzeit besteht die Fachschaft Informatik der Holzkamp-Gesamtschule aus z w e i Lehrkräften, denen zwei Computerräume mit 14 bzw. 29 Computerarbeitsplätzen und ein Selbstlernzentrum mit 4 Plätzen zur Verfügung stehen. Zusätzlich sind die Klas- sen- und Fachräume mit Computeranschlussmöglichkeiten, Beamern u.ä. ausgestat- tet. 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 in den schulischen Räumen zum Zugriff auf ihre eigenen Daten, zur Recherche im Internet oder zur Bearbeitung schulischer Aufga- ben verwenden können.
Der Unterricht erfolgt im 45-Minuten-Takt. Die Kursblockung sieht, wenn organisatorisch machbar, 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

Einführungsphase	
Unterrichtsvorhaben EF-I	**Unterrichtsvorhaben EF-II**
Thema: *Einführung in die Nutzung von Informatiksystemen in grundlegende Begrifflichkeiten*	**Thema:** *Grundlagen der objektorientierten Analyse, Modellierung und Implementierung anhand von statischen Grafikszenen und/oder Rollenspielen*
Zentrale Kompetenzen: Argumentieren Darstellen und Interpretieren Kommunizieren und Kooperieren	**Zentrale Kompetenzen:** Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren
Inhaltsfelder: Informatiksysteme Informatik, Mensch und Gesellschaft	**Inhaltsfelder:** Daten und ihre Strukturierung Formale Sprachen und Automaten
Inhaltliche Schwerpunkte: Einzelrechner Dateisystem Internet Einsatz von Informatiksystemen	**Inhaltliche Schwerpunkte:** Objekte und Klassen Syntax und Semantik einer Programmiersprache
Zeitbedarf: 6 Stunden	**Zeitbedarf:** 8 Stunden

Einführungsphase	
Unterrichtsvorhaben EF-III	**Unterrichtsvorhaben EF-IV**
Thema: *Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen in Java anhand von einfachen Animationen*	**Thema:** *Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand von grafischen Spielen und Simulationen*
Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Kommunizieren und Kooperieren	**Zentrale Kompetenzen:** Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren
Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Formale Sprachen und Automaten	**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	**Inhaltliche Schwerpunkte:** Objekte und Klassen Syntax und Semantik einer Programmiersprache Analyse, Entwurf und Implementierung einfacher Algorithmen
Zeitbedarf: 18 Stunden	**Zeitbedarf: 1**8 Stunden

Einführungsphase	
Unterrichtsvorhaben EF-V	**Unterrichtsvorhaben EF-VI**
Thema: *Such- und Sortieralgorithmen anhand kontextbezogener Beispiele*	**Thema:** *Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes*
Zentrale Kompetenzen: Argumentieren Modellieren Darstellen und Interpretieren Kommunizieren und Kooperieren	**Zentrale Kompetenzen:** Argumentieren Darstellen und Interpretieren Kommunizieren und Kooperieren
Inhaltsfelder: Algorithmen	**Inhaltsfelder:** Informatik, Mensch und Gesellschaft Informatiksysteme
Inhaltliche Schwerpunkte: Algorithmen zum Suchen und Sortieren Analyse, Entwurf und Implementierung einfacher Algorithmen	**Inhaltliche Schwerpunkte:** Wirkungen der Automatisierung Geschichte der automatischen Datenverarbeitung Digitalisierung
Zeitbedarf: 8 Stunden	**Zeitbedarf:** 6 Stunden

Einführungsphase	
Unterrichtsvorhaben EF-VII	
Thema: *Größeres Programmierprojekt*	
Zentrale Kompetenzen: Argumentieren Modellieren Darstellen und Interpretieren Kommunizieren und Kooperieren ggf. Implementieren	
Inhaltsfelder: Daten und Strukturierung Algorithmen ggf. Informatik, Mensch und Gesellschaft	
Inhaltliche Schwerpunkte: Analyse von realen Softwareprojekten Analyse und Entwurf von Lasten- und Pflichtenheften ggf. Implementation, Spielsucht	
Zeitbedarf: 10 Stunden	

II) Qualifikationsphase (Q1 und Q2) – GRUNDKURS

Qualifikationsphase 1

Unterrichtsvorhaben Q1-I	*Unterrichtsvorhaben Q1-II*
Thema: *Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung* **Zentrale Kompetenzen:** Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren **Inhaltsfelder:** Daten und ihre Strukturierung Algorithmen Formale Sprachen und Automaten Informatiksysteme **Inhaltliche Schwerpunkte:** Objekte und Klassen Analyse, Entwurf und Implementierung von Algorithmen Syntax und Semantik einer Programmiersprache Nutzung von Informatiksystemen **Zeitbedarf**: 8 Stunden	**Thema:** *Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen* **Zentrale Kompetenzen:** Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren **Inhaltsfelder:** Daten und ihre Strukturierung Algorithmen Formale Sprachen und Automaten **Inhaltliche Schwerpunkte:** Objekte und Klassen Analyse, Entwurf und Implementierung von Algorithmen Algorithmen in ausgewählten informatischen Kontexten Syntax und Semantik einer Programmiersprache **Zeitbedarf**: 20 Stunden

Qualifikationsphase 1	
Unterrichtsvorhaben Q1-III **Thema:** *Suchen und Sortieren auf linearen Datenstrukturen* **Zentrale Kompetenzen:** Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren **Inhaltsfelder:** Algorithmen Formale Sprachen und Automaten **Inhaltliche Schwerpunkte:** Analyse, Entwurf und Implementierung von Algorithmen Algorithmen in ausgewählten informatischen Kontexten Syntax und Semantik einer Programmiersprache **Zeitbedarf**: 16 Stunden	*Unterrichtsvorhaben Q1-IV* **Thema:** *Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten* **Zentrale Kompetenzen:** Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren **Inhaltsfelder:** Daten und ihre Strukturierung Algorithmen Formale Sprachen und Automaten Informatik, Mensch und Gesellschaft **Inhaltliche Schwerpunkte:** Datenbanken Algorithmen in ausgewählten informatischen Kontexten Syntax und Semantik einer Programmiersprache Sicherheit **Zeitbedarf**: 20 Stunden

Qualifikationsphase 1	
Unterrichtsvorhaben Q1-V **Thema:** *Sicherheit und Datenschutz in Netzstrukturen* **Zentrale Kompetenzen:** Argumentieren Darstellen und Interpretieren Kommunizieren und Kooperieren **Inhaltsfelder:** Informatiksysteme Informatik, Mensch und Gesellschaft **Inhaltliche Schwerpunkte:** Einzelrechner und Rechnernetzwerke Sicherheit Nutzung von Informatiksystemen, Wirkungen der Automatisierung **Zeitbedarf**: 10 Stunden	
Summe Qualifikationsphase 1: 74 Stunden	

Qualifikationsphase 2	
Unterrichtsvorhaben Q2-I **Thema:** *Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen* **Zentrale Kompetenzen:** Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren **Inhaltsfelder:** Daten und ihre Strukturierung Algorithmen Formale Sprachen und Automaten **Inhaltliche Schwerpunkte:** Objekte und Klassen Analyse, Entwurf und Implementierung von Algorithmen Algorithmen in ausgewählten informatischen Kontexten Syntax und Semantik einer Programmiersprache **Zeitbedarf**: 24 Stunden	*Unterrichtsvorhaben Q2-II* **Thema:** *Endliche Automaten und formale Sprachen* **Zentrale Kompetenzen:** Argumentieren Modellieren Darstellen und Interpretieren Kommunizieren und Kooperieren **Inhaltsfelder:** Endliche Automaten und formale Sprachen **Inhaltliche Schwerpunkte:** Endliche Automaten Grammatiken regulärer Sprachen Möglichkeiten und Grenzen von Automaten und formalen Sprachen **Zeitbedarf**: 20 Stunden

Qualifikationsphase 2	
<u>Unterrichtsvorhaben Q2-III</u> **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:** 12 Stunden	
Summe Qualifikationsphase 2: 56 Stunden	

2.1.2 Konkretisierte Unterrichtsvorhaben

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

In der Einführungsphase wird die didaktisierte Entwicklungsumgebung Greenfoot verwendet.

Für die Entwicklungsumgebung Greenfoot stehen Schulbücher für die SuS und die Lehrkraft zur Verfügung (Schöningh-Verlag: Informatik I). Die plattformunabhängigen Installationsdateien und Dokumentationen der Entwicklungsumgebung stehen Intenet zum kostenlosen Download zur Verfügung.

In der Qualifikationsphase werden die Unterrichtsvorhaben unter Berücksichtigung der Vorgaben für das Zentralabitur Informatik in NRW konkretisiert und in BlueJ bzw. mithilfe des JavaEditors (v.a. für grafische Oberflächen) umgesetzt.

Die Abiturvorgaben sind zu beziehen unter der Adresse

https://www.standardsicherung.schulministerium.nrw.de/cms/zentralabitur-gost/faecher/fach.php?fach=15
(Link aktualisiert am: 07. 12. 2022)

I Einführungsphase

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

Die Schülerinnen und Schüler

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

Unterrichtsvorhaben EF-I

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

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

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

Zunächst wird auf den Begriff der Information und die Bedeutung der Informatik eingegangen. Im Anschluss werden die Grundideen der Teilbereiche der Informatik erarbeitet und einfache Beispiele aus den Bereichen Technische Informatik (Know-How-Computer), Theoretische Informatik (Kürzeste Wege), Praktische Informatik (Sortierverfahren), Informatik und Gesellschaft (Gläserne Kunden) und Kryptographie (Caesar-Verschlüsselung) bearbeitet.

Material: Schöningh-Schulbuch: Informatik I, 2. Auflage, S. 10-14

Zeitbedarf: 6 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Information, deren Kodierung und Speicherung a. Informatik als Wissenschaft der Verarbeitung von Informationen b. Darstellung von Informationen in Schrift, Bild und Ton c. Speichern von Daten mit informatischen Systemen am Beispiel der Schulrechner d. Vereinbarung von Richtlinien zur Datenspeicherung auf den Schulrechnern (z.B. Ordnerstruktur, Dateibezeichner usw.)	Die Schülerinnen und Schüler beschreiben und erläutern die Begriffe Information und Informatik (A), nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D), kooperieren in Gruppen- und Partnerarbeit und erarbeiten neue Inhalte gemeinsam (K), präsentieren ihre Arbeitsergebnisse (K).	*z.B.* Textkodierung Kodierung und Dekodierung von Texten mit unbekannten Zeichensätzen (z.B. Wingdings) *z.B.* Bildkodierung Kodierung von Bildinformationen in Raster- und Vektorgrafiken *z.B.* Informationstexte zu Information und Informatik
2. Teilbereiche der Informatik a. Know-How-Computer als Beispiel für die Technische Informatik b. Kürzeste Wege suchen als Beispiel für die Theoretische Informatik c. Spielerische Einführung in den BubbleSort-Algorithmus als Beispiel für die Praktische Informatik d. Datenspeicherung am Beispiel Payback als Beispiel für den Bereich Informatik und Gesellschaft e. (De-) Codierung mit der Caesar-Scheibe als Beispiel für Kryptographie		*Beispiel*: Durchführung eines Gruppenpuzzles, in dem die SuS in Kleingruppen jeweils einen Teilbereich erarbeiten und den Mitschülern anschließend präsentieren. Im Anschluss erfolgt dann eine Sicherung der Ergebnisse *Beispiel:* Durchführung eines Stationenlernens, in dem die SuS selbstständig wählen dürfen, welche Teilbereiche sie sich detaillierter ansehen wollen. Im Anschluss erfolgt dann eine Vorstellung/Sicherung der Ergebnisse *Material:* Schöningh-Schulbuch: Informatik I, 2. Auflage, S. 10-14 & Caesar-Scheiben (Sammlung)

Unterrichtsvorhaben EF-II

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

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 Greenfoot begonnen. Anhand vorgegebener Klassen und Objekte wird spielerisch der Umgang mit Greenfoot eingeübt. In diesem Zusammenhang werden die Methoden und Attribute der Klassen und Objekte analysiert, bevor in die einfache Programmierung eingestiegen wird.

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
1. Identifikation von Objekten a. Am Beispiel eines lebensweltnahen Beispiels werden Objekte im Sinne der Objektorientierten Modellierung eingeführt. b. Objekte werden mit Objektkarten visualisiert und mit sinnvollen Attributen und „Fähigkeiten“, d.h. Methoden versehen. c. Manche Objekte sind prinzipiell typgleich und werden so zu einer Objektsorte bzw. Objektklasse zusammengefasst. d. Vertiefung: Modellierung weiterer Beispiele ähnlichen Musters	Die Schülerinnen und Schüler 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).	*Beispiel:* Pferde Schülerinnen und Schüler betrachten eine Pferdewiese als Menge gleichartiger Objekte, die in einer Klasse mit Attributen und Methoden zusammengefasst werden können. *Materialien:* Schöningh-Schulbuch: Informatik I, 2. Auflage, S. 14 ff.
2. Analyse von Klassen didaktischer Lernumgebungen a. Objektorientierte Programmierung als modularisiertes Vorgehen (Entwicklung von Problemlösungen auf Grundlage vorhandener Klassen) b. Teilanalyse der Klassen der didaktischen Lernumgebungen Greenfoot c. Verwendung vorgegebener Methoden zum Kennenlernen der Lernumgebung		*Materialien*: Schöning-Schulbuch: Informatik 1, 2. Auflage, S. 19 ff.
3. Implementierung linearer Bewegungsabläufe in vorgegebenen Methoden a. Grundaufbau einer Java-Klasse b. Aufbau von Objekten c. Entwurf einfacher, linearer Bewegungsabläufe d. Implementation linearer Bewegungsabläufe		*Beispiele:* siehe Material *Materialien:* Schöning-Schulbuch: Informatik 1, 2. Auflage, S. 23 ff.

Unterrichtsvorhaben EF-III

Thema: Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen in Java anhand von einfachen Animationen

Leitfragen: *Wie findet Robby/Rover selbstständig seinen Weg durch ein Labyrinth?*

Vorhabenbezogene Konkretisierung:
Der Schwerpunkt dieses Unterrichtsvorhabens liegt auf der Entwicklung eines größeren Projekts, in dem Robby/Rover seinen Weg innerhalb eines beliebigen Labyrinths finden muss.

Hierfür lernen die SuS zunächst bedingte Anweisungen (if-Abfrage), Vorprüfende Wiederholung (while-Schleife) und die Zählschleife (for-Schleife) anhand kleinerer Miniprojekte kennen. Im Anschluss an die Einführung der Grundstrukturen in Java werden kurz die logischen Verknüpfungen zur Nutzung im Rahmen des Großprojekts eingeführt.

Die Grundstrukturen nutzen die SuS dann, um ein Programm zu schreiben, in dem Robby/Rover seinen Weg durch ein Labyrinth findet.

Zum Entwurf der Strukturen und einer Idee zur Lösung des Labyrinth-Problems wird die Darstellung in einem Programm-Ablauf-Plan verwendet.
Zeitbedarf: 18 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Einführung der Grundstrukturen in Java a. Analyse (mithilfe des PAP) und Implementation der Grundlagen der bedingten Anweisung anhand eines einfachen Problems b. Analyse (mithilfe des PAP) und Implementation der Grundlagen der vorprüfenden Wiederholung anhand eines einfachen Problems c. Analyse (mithilfe des PAP) und Implementation der Grundlagen der Zählschleife anhand eines einfachen Problems	Die Schülerinnen und Schüler erläutern und begründen methodische Vorgehensweisen, Entwurfs- und Implementierungsentscheidungen (A), konstruieren zu kontextbezogenen Problemstellungen informatische Modelle (M), implementieren auf Grundlage von Modellen ein Programmierprojekt in Greenfoot (I), testen und korrigieren ihr Programmierprojekt in Greenfoot (I), stellen informatische Abläufe in Programm-Ablauf-Plänen (PAP) dar (D), überführen gegebene grafische Darstellungen (PAP) in Java-Quellcode (D), verwenden Fachausdrücke (Bedingte Anweisung, Vorprüfende Wiederholung, Zählschleife) (K), kommunizieren und kooperieren in Gruppen (K), präsentieren ihre Gruppenergebnisse aus den Expertengruppen in Stammgruppen (K), präsentieren ihre Entwürfe und ihren Programmiercode (K)	*Beispiel:* Durchführung eines Gruppenpuzzles (vgl. Materialsammlung), in dem die SuS in Kleingruppen die einzelnen Grundstrukturen erarbeiten. Die neuen Kenntnisse werden dann in einer zweiten Phase im Sinne des Gruppenpuzzles an die anderen Gruppenmitglieder vermittelt. *Material:* Materialien zum Gruppenpuzzle in der Materialsammlung der Informatik; Schöningh-Schulbuch, 2. Auflage, S. 36-56
2. Durchführung eines Programmierprojekts a. Entwurf eines Programms zur Anwendung der Grundstrukturen in Greenfoot b. Einführung, Analyse und Implementation der logischen Verknüpfungen UND, ODER und NICHT zur Nutzung im Zusammenhang mit den Grundstrukturen in Java c. Implementation des Programms zur Anwendung der Grundstrukturen und logischen Verknüpfungen in Greenfoot		*Beispiel:* In den gemischten Gruppen aus der ersten Sequenz kann ein größeres Programmierprojekt durchgeführt werden, in dem die einzelnen Grundstrukturen genutzt werden müssen. Eine Möglichkeit hierfür besteht in der Wegsuche durch ein Labyrinth. Zum Entwurf der Problemlösung kann ein PAP verwendet werden. Die logischen Verknüpfungen werden im Zusammenhang mit dem Projekt eingeführt *Material:* Materialien zum Gruppenpuzzle in der Materialsammlung der Informatik; Schöningh-Schulbuch, 2. Auflage, S. 63 ff.

Unterrichtsvorhaben EF-IV

Thema: Modellierung und Implementierung von Methoden, Attributen und Klassen- und Objektbeziehungen anhand von grafischen Spielen

Leitfrage: *Wie lassen sich Methoden, Attribute und komplexere Beziehungen zwischen Objekten und Klassen realisieren?*

Vorhabenbezogene Konkretisierung:
Dieses Unterrichtsvorhaben beschäftigt sich zuerst mit der Initialisierung und Deklaration, sowie Wertzuweisung von Attributen und den Methodentypen Aufruf- und Anfrage-Methode, Konstruktoren sowie Parametern. Anschließend werden Beziehungen zwischen Klassen in Form von Vererbung und KENNT-Beziehung thematisiert.

Zeitbedarf: 18 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Entwurf und Implementation von Attributen und Variablen in Java a. Bedeutung und Analyse von Attributen in Java b. Einführung von Datentypen in Java c. Deklaration, Initialisierung und Wertzuweisungen von Attributen in Java	Die Schülerinnen und Schüler Erläutern und begründen methodische Vorgehensweisen, Entwurfs- und Implementierungsentscheidungen (A), analysieren und erläutern eine objektorientierte Modellierung (A), stellen die Beziehung zwischen Klassen grafisch dar (M), ermitteln bei der Analyse einfacher Problemstellungen die Eigenschaften und Operationen der Klasse, sowie ihre Beziehungen zu anderen Klassen (M), modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen zu (M), ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M), modellieren Klassen unter Verwendung von Vererbung (M), implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), testen Programme schrittweise anhand von Beispielen (I), stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), überführen gegebene Klassendiagramme in Quellcode (D), verwenden Fachausdrücke bei der Kommunikation über informatische Sachverhalte (K), kommunizieren und kooperieren in Gruppen- und Partnerarbeit, präsentieren Arbeitsabläufe und –ergebnisse (K).	*Beispiele:* Robby soll verschiedene Eigenschaften (z.B. Gewicht, Energie, Anzahl Akkus) besitzen. Diese Attribute sollen in verschiedenen bereits bekannten Methoden verwendet werden. *Material:* Schöningh-Schulbuch: Informatik 1, 2. Auflage, S. 77 ff.
2. Entwurf und Implementierung verschiedener Methodentypen in Java a. Entwurf und Implementierung von Aufruf- und Anfragemethoden in Java b. Entwurf und Implementierung von Parametern in Methoden in Java c. Einführung in die Bedeutung von Konstruktoren und Implementation eines Konstruktors mit und ohne Parameter		*Beispiele:* Robby soll mithilfe einer selbstprogrammierten Anfragemethode überprüfen, ob hinter ihm eine Wand steht und ansonsten rückwärts laufen (Aufruf-Methode). *Beispiel Parameter:* Robby soll immer eine bestimmte Anzahl an Schritte vorwärts gehen, welche ihm mithilfe eines Parameters mitgeteilt wird. *Beispiel Konstruktor:* Robby erhält mithilfe des Konstruktors vorgegebene Startwerte (z.B. Energie, Name, Schraubenanzahl) *Material:* Schöningh-Schulbuch: Informatik 1, 2. Auflage, S. 85 ff.
3. Beziehungen zwischen Klassen a. Wiederholung der Modellierung von Klassendiagrammen b. Einführung und Modellierung der Beziehung *Vererbung* und *Kennt-Beziehung* zwischen Klassen c. Implementation neuer Klassen und Herstellung von Beziehungen zwischen Klassen in Java (inkl. Erzeugung von Objekten in Java-Code) d. Einführung des Geheimnisprinzips und der Datenkapselung		*Beispiel:* Entwurf der Klasse Ölfleck und Erarbeitung der Beziehungen zu anderen Klassen. Implementation der Klasse Ölfleck. *Material:* Schöning-Schulbuch: Informatik 1, 2. Auflage, S. 100 ff. Analoge Beispiele für Rover

Unterrichtsvorhaben EF-V

Thema: Such- und Sortieralgorithmen anhand kontextbezogener Beispiele

Leitfragen: *Wie können Objekte bzw. Daten effizient sortiert werden, so dass eine schnelle Suche möglich wird?*

Vorhabenbezogene Konkretisierung:
Dieses Unterrichtsvorhaben beschäftigt sich mit der Erarbeitung von Such- und Sortieralgorithmen. Der Schwerpunkt des Vorhabens liegt dabei auf den Algorithmen selbst und nicht auf deren Implementierung in einer Programmiersprache, auf die in diesem Vorhaben vollständig verzichtet werden soll.

Zunächst erarbeiten die Schülerinnen und Schüler mögliche Einsatzszenarien für Such- und Sortieralgorithmen, um sich der Bedeutung einer effizienten Lösung dieser Probleme bewusst zu werden. Anschließend werden Strategien zur Sortierung mit Hilfe eines explorativen Spiels von den Schülerinnen und Schülern selbst erarbeitet und hinsichtlich der Anzahl notwendiger Vergleiche auf ihre Effizienz untersucht.

Daran anschließend werden die erarbeiteten Strategien systematisiert und im Pseudocode notiert. Die Schülerinnen und Schüler sollen auf diese Weise das *Sortieren durch Vertauschen*, das *Sortieren durch Auswählen* und mindestens einen weiteren Sortieralgorithmus, kennen lernen.

Zeitbedarf: 8 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Explorative Erarbeitung eines Sortierverfahrens a. Sortierprobleme im Kontext informatischer Systeme und im Alltag (z.B. Dateisortierung, Tabellenkalkulation, Telefonbuch, Bundesligatabelle, usw.) b. Vergleich zweier Elemente als Grundlage eines Sortieralgorithmus c. Erarbeitung eines Sortieralgorithmus durch die Schülerinnen und Schüler	Die Schülerinnen und Schüler beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeit und Speicherplatzbedarf (A), entwerfen einen weiteren Algorithmus zum Sortieren (M), analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D).	*Beispiel:* Sortieren mit Waage Die Schülerinnen und Schüler bekommen die Aufgabe, kleine, optisch identische Kunststoffbehälter aufsteigend nach ihrem Gewicht zu sortieren. Dazu steht ihnen eine Balkenwaage zur Verfügung, mit deren Hilfe sie das Gewicht zweier Behälter vergleichen können. *Beispiel:* Sortieren mit Spielkarten Die SuS erhalten Spielkarten, die sie unter bestimmten Regeln sortieren sollen. *Materialien:* Computer science unplugged – Sorting Algorithms, URL: https://youtu.be/cVMKXKoGu_Y abgerufen: 27.07.2023 Schöningh-Schulbuch: Informatik 2, 1. Auflage, S. 94 ff.
2. Systematisierung von Algorithmen und Effizienzbetrachtungen a. Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode (auf jeden Fall: Sortieren durch Vertauschen, Sortieren durch Auswählen) b. Anwendung von Sortieralgorithmen auf verschiedene Beispiele c. Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche d. Variante des Sortierens durch Auswählen (Nutzung eines einzigen oder zweier Felder bzw. lediglich eines einzigen zusätzlichen Ablageplatzes oder mehrerer neuer Ablageplätze) e. Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des Speicherplatzbedarfs f. Analyse des weiteren Sortieralgorithmus (sofern nicht in Sequenz 1 und 2 bereits geschehen)		*Beispiele*: Sortieren durch Auswählen, Sortieren durch Vertauschen, Quicksort Quicksort ist als Beispiel für einen Algorithmus nach dem Prinzip *Teile und Herrsche* gut zu behandeln. Kenntnisse in rekursiver Programmierung sind nicht erforderlich, da eine Implementierung nicht angestrebt wird. *Materialien:* Computer science unplugged – Sorting Algorithms, URL: https://youtu.be/cVMKXKoGu_Y abgerufen: 27.07.2023

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:
Im folgenden Unterrichtsvorhaben sollen Schülerinnen und Schüler selbstständig informatische Themenbereiche aus dem Kontext der Geschichte der Datenverarbeitung und insbesondere den daraus sich ergebenen 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.

Zeitbedarf: 6 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	**Zu entwickelnde Kompetenzen**	**Beispiele, Medien, Materialien**
1. Selbstständige Erarbeitung von Themen durch die Schülerinnen und Schüler a. Mögliche Themen zur Erarbeitung in Kleingruppen: „Geschichte der Digitalisierung: Vom Morsen zum modernen Digitalcomputer“ „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“ „Datenskandale – Wie unsere Daten geschützt werden“ (Bundesdatenschutzgesetz u.ä.) b. Vorstellung und Diskussion durch Schülerinnen und Schüler	Die Schülerinnen und Schüler bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (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).	*Beispiel:* Ausstellung zu informatischen Themen 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. *Materialien:* Schülerinnen und Schüler recherchieren selbstständig im Internet, in der Schulbibliothek, in öffentlichen Bibliotheken, usw.

Unterrichtsvorhaben EF-VII

Thema: Größeres Programmierprojekt

Leitfrage: *Wie arbeiten Softwareunternehmen mit Kunden zusammen?*

Vorhabenbezogene Konkretisierung:
Das folgende Unterrichtsvorhaben stellt den Abschluss der Einführungsphase dar.

Konkret wird hier auf Basis eines Lastenheftes, das von einem fiktiven Unternehmen gestellt worden ist, ein Pflichtenheft als Vertragsangebot erarbeitet.

Zeitbedarf: 10 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Einführung in Lastenheft und Pflichtenheft a. Die Begriffe Lastenheft und Pflichtenheft werden erarbeitet b. Anhand kleinerer Beispiele wird die Erstellung eines Pflichtenhefts geübt	Die Schülerinnen und Schüler Erläutern und begründen methodische Vorgehensweisen bei der Erstellung eines Pflichtenhefts (A), analysieren und erläutern Lasten- und Pflichtenhefte (A), beurteilen die Angemessenheit von Pflichtenheften auf Basis vorgegebener Lastenhefte (A), konstruieren auf Basis vorgegebener Lastenhefte und Anforderungen ein Pflichtenheft (M), interpretieren Anforderungen des Lastenhefts und erläutern Beziehungen zwischen verschiedenen Anforderungen des Lastenhefts (D), überführen gegebene textuel- le und grafische Darstellun- gen des Lastenhefts auf das Pflichtenheft (D), verwenden Fachausdrücke bei der Kommunikation über informatische Sachverhalte (K), kommunizieren und kooperieren in Gruppen (K), präsentieren ihr Pflichtenheft im schulischen und ggf. außerschulischen Rahmen beim Kooperationspartner vor Ort (K). cke, modellieren auf Grundlage des Pflichtenhefts Klassendiagramme (M), implementieren auf Grundlage des Pflichtenhefts und Klassendiagramms ein Spiel (I), testen und korrigieren ihr Computerspiel (I).	*Beispiel:* Kleinere Beispiele aus der bekannten Entwicklungsumgebung werden den SuS in Form eines Lastenhefts vorgegeben. Diese müssen die SuS in ein Pflichtenheft überführen.
2. Erstellung eines Pflichtenhefts		*Material:* Bereitstellung des Lastenhefts und der Anforderungen durch das Softwareunternehmen
3. Modellierung und Implementation eines Computerspiels (optional) a. Modellierung eines Klassendiagramms auf Basis des erstellten Pflichtenhefts b. Implementation des Computerspiels auf Basis von Pflichtenheft und Klassendiagramm		

II) Qualifikationsphase

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

Die Schülerinnen und Schüler

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

Unterrichtsvorhaben Q1-I:

Thema: Wiederholung der objektorientierten Modellierung und Programmierung

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

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

Die Schülerinnen und Schülern erläutern und modifizieren den ersten Entwurf und modellieren sowie implementieren weitere Klassen und Methoden für eine entsprechende Anwendung. Klassen und ihre Beziehungen werden in einem Implementationsdiagramm dargestellt. Dabei werden Sichtbarkeitsbereiche zugeordnet. Exemplarisch wird eine Klasse dokumentiert. Der Nachrichtenaustausch zwischen verschiedenen Objekten wird verdeutlicht, indem die Kommunikation zwischen zwei ausgewählten Objekten grafisch dargestellt wird. In diesem Zusammenhang wird das Nachrichtenkonzept der objektorientierten Programmierung wiederholt.

Zeitbedarf: 8 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Wiederholung und Erwei- terung der objektorientierten Modellierung und Programmierung durch Analyse und Erweiterung eines kontextbezogenen Beispiels (a) Analyse der Problem- stellung (b) Analyse der Modellie- rung (Implementatio ns- diagramm) (c) Erweiterung der Modellierung im Implementati- onsdiagramm (Vererbung, abstrakte Klasse) (d) Kommunikatio n zwi- schen mindestens zwei Objekten (grafische Darstellung) (e) Dokumentation von Klassen (f) Implementieru ng der Anwendung oder von Teilen der Anwendung	Die Schülerinnen und Schüler analysieren und erläutern objektorientierte Model- lierungen (A), beurteilen die syntaktische Korrektheit und die Funktio- nalität von Programmen (A), modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbe- reiche zu (M), modellieren abstrakte und nicht abstrakte Klassen unter Ver- wendung von Vererbung durch Spezialisieren und Generali- sieren (M), implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klas- senbibliotheken (I), nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), wenden eine didaktisch orientierte Entwicklungsumge- bung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatik- systemen an (I), interpretieren Fehlermeldungen und korri- gieren den Quellcode (I), stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), dokumentieren Klassen (D), stellen die Kommunikation zwischen Objekten grafisch dar (D).	*Beispiel:* Wetthuepfen Für ein Wetthüpfen zwischen einem Hasen, einem Hund und einem Vogel werden die Tiere gezeichnet. Alle Tiere springen wiederholt nach links. Die Höhe und Weite jedes Hüpfers ist zufällig. Evtl. marschieren sie anschließend hintereinander her. oder *Beispiel*: Tannenbaum Ein Tannenbaum soll mit verschiedenen Arten von Schmuckstücken versehen werden, die durch unter- schiedliche geometrische Ob- jekte dargestellt werden. Es gibt Kugeln, Päckchen in der Form von Würfeln und Zuckerringe in Form von Toren. Ein Prototyp, der bereits mit Kugeln geschmückt werden kann, kann zur Verfügung gestellt werden. Da alle Schmuckstücke über die Funktion des Auf- und Ab- schmückens verfügen sollen, liegt es nahe, dass entspre- chende Methoden in einer gemeinsamen Oberklasse realisiert werden. *Materialien:* Ergänzungsmaterialien zum Lehrplannavigator Unter- richtsvorhaben Q1.1- Wiederholung (Download Q1-I.1)

Unterrichtsvorhaben Q1-II:

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

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

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

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse `Queue` (a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen (b) Erarbeitung der Funktionalität der Klasse `Queue` (c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse `Queue`	Die Schülerinnen und Schüler erläutern Operationen dynamischer (linearer oder nichtlinearer) Datenstrukturen (A), analysieren und erläutern Algorithmen und Programme (A), beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaf ten, ihre Operationen und ihre Beziehungen (M), modifizieren Algorithmen und Programme (I), implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), interpretieren Fehlermeldungen und korrigieren den Quellcode (I), testen Programme systematisch anhand von Beispielen (I), stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D).	*Beispiel:* Patientenwarteschlange (jeder kennt seinen Nachfolger bzw. alternativ: seinen Vorgänger) Sobald ein Patient in einer Arztpraxis eintrifft, werden sein Name und seine Krankenkasse erfasst. Die Verwaltung der Patientenwarteschlange geschieht über eine Klasse, die hier als Wartezimmer bezeichnet wird. Wesentliche Operationen sind das „Hinzufügen“ eines Patienten und das „Entfernen“ eines Patienten, wenn er zur Behandlung gerufen wird. Die Simulationsanwendung stellt eine GUI zur Verfügung, legt ein Wartezimmer an und steuert die Abläufe. Wesentlicher Aspekt des Projektes ist die Modellierung des Wartezimmers mit Hilfe der Klasse Queue. Anschließend wird der Funktionsumfang der Anwendung erweitert: Patienten können sich zusätzlich in die Warteschlange zum Blutdruckmessen einreihen. Objekte werden von zwei Schlangen verwaltet. *Materialien:* Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.2 – Warteschlange (Download Q1-II.1)

2. Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse `Stack` (a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen (b) Erarbeitung der Funktionalität der Klasse `Stack` (c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse `Stack`		*Beispiel:* Heftstapel In einem Heftstapel soll das Heft einer Schülerin gefunden werden. oder *Beispiel*: Kisten stapeln In einem Stapel nummerierter Kisten soll eine bestimmte Kiste gefunden und an einen Kunden geliefert werden. Dazu müssen Kisten auf verschiedene Stapel gestapelt und wieder zurückgestellt werden.
3. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse `List` (a) Erarbeitung der Vorteile der Klasse `List` im Gegensatz zu den bereits bekannten linearen Strukturen (b) Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse `List`.		*Beispiel*: Abfahrtslauf Bei einem Abfahrtslauf kommen die Skifahrer nacheinander an und werden nach ihrer Zeit in eine Rangliste eingeordnet. Diese Rangliste wird in einer Anzeige ausgegeben. Ankommende Abfahrer müssen an jeder Stelle der Struktur, nicht nur am Ende oder Anfang, eingefügt werden können. *Materialien:* Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.2 - Listen (Download Q1-II.2)

4. Vertiefung - Anwendungen von Listen, Stapeln oder Schlangen in mindestens einem weiteren Kontext		*Beispiel:* Skispringen Ein Skispringen hat folgenden Ablauf: Nach dem Sprung erhält der Springer eine Punktzahl und wird nach dieser Punktzahl in eine Rang-liste eingeordnet. Die besten 30 Springer qualifizieren sich für den zweiten Durchgang. Sie starten in umgekehrter Reihenfolge gegenüber der Platzierung auf der Rangliste. Nach dem Sprung erhält der Springer wiederum eine Punktzahl und wird nach der Gesamt-punktzahl aus beiden Durchgängen in die endgültige Rangliste eingeordnet. *Beispiel*: Terme in Postfix- Notation Die sog. UPN (*Umgekehrt- Polnische-Notation*) bzw. *Postfix-Notation* eines Terms setzt den Operator hinter die Operanden. Um einen Term aus der gewohnten Infixschreibweise in einen Term in UPN umzuwandeln oder um den Wert des Terms zu berechnen, kann ein Stack verwendet werden. *Beispiel*: Rangierbahnhof Auf einem Güterbahnhof gibt es drei Gleise, die nur zu einer Seite offen sind. Wagons kön- nen also von einer Seite auf das Gleis fahren und nur rückwärts wieder hinausfahren. Die Wagons tragen Nummern, wobei die Nummer jedoch erst eingesehen werden kann, wenn der Wagon der vorderste an der offenen Gleisseite ist. (Zwischen den Wagons herumzu-turnen, um die anderen Wagonnummern zu lesen, wäre zu gefährlich.) Zunächst stehen alle Wagons unsortiert auf einem Gleis. Ziel ist es, alle Wa- gons in ein anderes Gleis zu fahren, so dass dort die Nummern der Wagons vom Glei- sende aus aufsteigend in richtiger Reihenfolge sind. Zusätzlich zu diesen beiden Gleisen gibt es ein Abstellgleis, das zum Rangieren benutzt werden kann. *Beispiel*: Autos an einer Ampel zur Zufahrtsregelung Es soll eine Ampel zur Zufahrts regelung in Java simuliert wer den. An einem geradlinigen, senkrecht von unten nach oben verlaufenden Straßenstück, das von Autos nur einspurig in eine Richtung befahren werden kann, ist ein Haltepunkt mar kiert, an dem die Ampel steht. Bei einem Klick auf eine Schalt fläche mit der Aufschrift „Heran fahren“ soll ein neues Auto an den Haltepunkt heranfahren bzw. bis an das letzte Auto, das vor dem Haltepunkt wartet. Grünphasen der Ampel werden durch einen Klick auf eine Schaltfläche mit der Aufschrift „Weiterfahren“ simuliert. In jeder Grünphase darf jeweils nur ein Auto weiterfahren. Die an- deren Autos rücken nach. *Materialien:* Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1-II.3 – Anwendungen für lineare Datenstrukturen (Download Q1-II.3)

Unterrichtsvorhaben Q1-III:

Thema: Suchen und Sortieren auf linearen Datenstrukturen

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

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

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

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

Zeitbedarf: 16 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompe-Tenzen	Beispiele, Medien, Materialien
1. Suchen von Daten in Listen und Arrays (a) Lineare Suche in Lis- ten und in Arrays (b) Binäre Suche in Ar- rays als Beispiel für rekursives Problemlö- sen (c) Untersuchung der beiden Suchverfahren hinsichtlich ihrer Effi- zienz (Laufzeitverhal-ten, Speicherbedarf)	Die Schülerinnen und Schü- ler analysieren und erläutern Algorithmen und Pro-gramme (A), beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A), entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M), modifizieren Algorithmen und Programme (I), implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I), nutzen die Syntax und Semantik einer Program-miersprache bei der Implementierung und zur Analyse von Programmen (I), interpretieren Fehlermel-dungen und korrigieren den Quellcode (I), testen Programme sys-tematisch anhand von Beispielen (I), stellen iterative und rekursive Algorithmen umgangs-sprachlich und grafisch dar (D).	*Beispiel:* Karteiverwaltung Für ein Adressverwaltungspro-gramm soll eine Methode zum Suchen einer Adresse ge-schrieben werden. oder *Beispiel*: Bundesjugendspie-le Die Teilnehmer an Bundesju-gendspielen nehmen an drei Disziplinen teil und erreichen dort Punktzahlen. Diese werden in einer Wettkampfkarte einge-tragen und an das Wettkampf-büro gegeben. Zur Vereinfa-chung sollte sich das Modell auf die drei Disziplinen „Lauf”, „Sprung“ und „Wurf“ beschrän-ken. Im Wettkampfbüro wird das Ergebnis erstellt. Das Programm soll dafür zunächst den Besten einer Disziplin heraussuchen können und später das gesamte Ergebnis nach gewissen Kriterien sortieren können. *Materialien:* Ergänzungsmaterialien zum Lehrplannavigator Unterrichts-vorhaben Q1.3 - Suchen und Sortieren (Download Q1-III.1)

2. Sortieren in Listen und Arrays - Entwicklung und Implementierung von iterativen und rekursiven Sortierverfahren (a) Entwicklung und Implementierung eines einfachen Sortierverfahrens für eine Liste (b) Implementierung eines einfachen Sortierverfahrens für ein Feld (c) Entwicklung eines rekursiven Sortierverfahren für ein Feld (z.B. Sortieren durch Mischen)		*Beispiel:* Karteiverwaltung (s.o.) oder *Beispiel*: Bundesjugendspiele (s.o.) *Materialien:* (s.o.)
3. Untersuchung der Effizienz der Sortierverfahren „Sortieren durch direktes Einfügen“ und „Quicksort“ auf linearen Listen (a) Grafische Veranschaulichung der Sortierverfahren (b) Untersuchung der Anzahl der Vergleichsoperationen und des Speicherbedarf bei beiden Sortierverfahren (c) Beurteilung der Effizienz der beiden Sortierverfahren		*Beispiel:* Karteiverwaltung (s.o.) oder *Beispiel*: Bundesjugendspiele (s.o.) *Materialien:* (s.o.)

Unterrichtsvorhaben Q1-IV:

Thema: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

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

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

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

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

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

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Nutzung von relationalen Datenbanken (a) Aufbau von Datenbanken und Grundbegriffe Entwicklung von Fragestellungen zur vorhandenen Datenbank Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, Datenbankschema (b) SQL-Abfragen Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL (SELECT (DISTINCT) …FROM, WHERE, AND, OR, NOT) auf einer Tabelle Analyse und Erarbeitung von SQL-Abfragen auf einer und mehrerer Tabelle zur Beantwortung der Fragestellungen (JOIN, UNION, AS, GROUP BY,ORDER BY, `ASC, DESC, COUNT, MAX, MIN, SUM`, Arithmetische Operatoren: `+, -, *, /, (…)`, Vergleichsoperatoren: `=, <>, >, <, >=, <=, LIKE, BETWEEN, IN, IS NULL`) (c) Vertiefung an einem weiteren Datenbank-beispiel	Die Schülerinnen und Schüler erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A), analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A), analysieren und erläutern eine Datenbankmodellierung (A), erläutern die Eigenschaften normalisierter Datenbankschemata (A), bestimmen Primär- und Sekundärschlüssel (M), ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M), modifizieren eine Datenbankmodel-lierung (M), modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M), bestimmen Primär- und Sekundärschlüssel (M), überführen Datenbankschemata in vorgegebene Normalformen (M), verwenden die Syntax und Semantik einer Datenbankabfrage-sprache, um Informationen aus einen Datenbanksystem zu extrahieren (I), ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D),	*Beispiel:* FitnessCenter FitnessCenter ist die Simulation einer Fitnes-Centers für den Informatik-Unterricht mit Webfrontends zur Verwaltung der Kunden, die an Kursen teilnehmen können. Außerdem ist es möglich direkt SQL-Abfragen einzu- geben. Es ist auch möglich, die Datenbank herunter zu laden und lokal zu installieren. Unter https://dokumentation.fitnesscenter.schule.de/ (abgerufen: 27.07.2023) findet man den Link zu dem FitnessCenter-System sowie nähere Informationen. Lesenswert ist auch die dort verlinkte „Dokumentation der Fallstudie" mit didaktischem Material, welches alternativ bzw. ergänzend zu der im Folgenden beschriebenen Durchführung verwendet werden kann.

2. Modellierung von relationalen Datenbanken (a)Entity-Relationship Diagramm Ermittlung von Entitäten, zugehörigen Attributen, Relationen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms Erläuterung und Modifizierung einer Datenbankmodellierung (b)Entwicklung einer Datenbank aus einem Datenbankentwurf Modellierung eines relationalen Datenbankschematas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln (c)Redundanz, Konsistenz und Normalformen Untersuchung einer Datenbank hinsichtlich Konsistenz und Redundanz in einer Anwendungssituation Überprüfung von Datenbankschemata hinsichtlich der 1. Bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten)	stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D), überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D).	*Beispiel:* Fahrradverleih Der Fahrradverleih *BTR (BikesToRent)* verleiht unterschiedliche Typen von Fahrrädern diverser Firmen an seine Kunden. Die Kunden sind bei *BTR* registriert (Name, Adresse, Telefon). *BTR* kennt von den Fahrradfirmen den Namen und die Telefonnummer. Kunden von *BTR* können CityBikes, Treckingräder und Mountainbikes ausleihen. *Beispiel*: Reederei Die Datenverwaltung einer Reederei soll in einem Datenbanksystem umgesetzt werden. Ausgehend von der Modellierung soll mit Hilfe eines ER-Modells und eines Datenbankschemas dieser erste Entwurf normalisiert und in einem Datenbanksystem umgesetzt werden. Es schließen sich diverse SQL-Abfragen an, wobei auf die Relationenalgebra eingegangen wird. *Beispiel:* Buchungssystem In dem Online-Buchungssystem einer Schule können die Lehrer Medienräume, Beamer, Lap- tops, Kameras, usw. für einen bestimmten Zeitpunkt buchen, der durch Datum und die Schulstunde festgelegt ist. Dazu ist die Datenbank zu modellieren, ggf. zu normalisieren und im Datenbanksystem umzusetzen. Weiter sollen sinnvolle Abfragen entwickelt werden. *Beispiel: Schulverwaltung* In einer Software werden die Schulhalbjahre, Jahrgangsstufen, Kurse, Klassen, Schüler, Lehrer und Noten einer Schule verwaltet. Man kann dann ablesen, dass z.B. Schüler X von Lehrer Y im 2. Halbjahr des Schuljahrs 2011/2012 in der Jahrgangsstufe 9 im Differen zierungsbereich im Fach Informatik die Note „sehr gut“ erhalten hat. Dazu ist die Datenbank zu modellieren, ggf. zu normalisieren und im Datenbanksystem umzusetzen. Weiter sollen sinnvolle Abfragen entwickelt werden und das Thema Datenschutz besprochen werden.

Unterrichtsvorhaben Q1-V:

Thema: Sicherheit und Datenschutz in Netzstrukturen

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

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

Zeitbedarf: 10 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Daten in Netzwerken und Sicherheitsaspekte in Netzen sowie beim Zugriff auf Datenbanken (a)Beschreibung eines Datenbankzugriffs im Netz anhand eines Anwendungskontextes und einer Client-Server-Struktur zur Klärung der Funktionsweise eines Datenbankzugriffs (b)Netztopologien als Grundlage von Client- Server-Strukturen und TCP/IP-Schichtenmodell als Beispiel für eine Paketübermittlung in einem Netz (c) Vertraulichkeit, Integrität, Authentizität in Netzwerken sowie symmetrische und asymmetrische kryptografische Verfahren (Cäsar-, Vigenère-, RSA-Verfahren) als Methoden Daten im Netz verschlüsselt zu übertragen	Die Schülerinnen und Schüler beschreiben und erläutern Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A), analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A), untersuchen und bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen, die Sicherheit von Informatiksystemen sowie die Einhaltung der Datenschutzbestimmungen und des Urheberrechts (A), untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A), nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D).	*Materialien:* Ergänzungsmaterialien zum Lehrplannavigator Unterrichts vorhaben, Verschlüsselung Q1.5 - Zugriff auf Daten in Netzwerken (Download Q1-V.1)
2. **Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht**		*Materialien:* *Ergänzungsmaterialien zum Lehrplannavigator Unterrichts- vorhaben Q1 5* - Datenschutz beim Videocenter, Materialblatt-Datenschutzgesetz (Download Q1-V.2)

Unterrichtsvorhaben Q2-I:

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

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

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

Unter anderem sollen die verschiedenen Baumtraversierungen (Pre-, Post- und Inorder) implementiert werden. Unterschiede bezüglich der Möglichkeit, den Baum anhand der Ausgabe der Bauminhalte via Pre-, In- oder Postorder-Traversierung zu rekonstruieren, werden dabei ebenfalls angesprochen, indem die fehlende Umkehrbarbeit der Zuordnung Binärbaum Inorder-Ausgabe an einem Beispiel verdeutlicht wird. Eine Tiefensuche wird verwendet, um einen in der Baumstruktur gespeicherten Inhalt zu suchen.

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

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

Zeitbedarf: 24 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Analyse von Baumstrukturen in verschiedenen Kontexten (a) Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit) (b) Aufbau und Darstellung von binären Bäumen an- hand von Baumstrukturen in verschiedenen Kontexten	Die Schülerinnen und Schüler erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), analysieren und erläutern Algorithmen und Programme (A), beurteilen die syntak- tische Korrektheit und die Funktionalität von Programmen (A), ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Verer bung durch Spezialisieren und Generalisieren (M), verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M), entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M),	*Beispiel:* Termbaum Der Aufbau von Termen wird mit Hilfe von binären Baumstrukturen verdeutlicht. oder *Beispiel*: Ahnenbaum Die binäre Baumstruktur ergibt sich daraus, dass jede Person genau einen Vater und eine Mutter hat. *Weitere Beispiele für Anwendungskontexte für binäre Bäume:* *Beispiel*: Suchbäume (zur sortierten Speicherung von Daten) Alle Inhalte, die nach einer Ordnung vor dem Inhalt im aktuellen Teilbaum ste- hen, sind in dessen linkem Teilbaum, alle die nach dem Inhalt im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.) *oder* *Beispiel*: Codierungsbäume für Codierungen, deren Alphabet aus genau zwei Zeichen besteht Morse hat Buchstaben als Folge von Punkten und Strichen codiert. Diese Codierungen können in einem Binärbaum dargestellt werden, so dass ein Übergang zum linken Teilbaum einem Punkt und ein Übergang zum rechten Teilbaum einem Strich entspricht. Wenn man im Gesamtbaum startet und durch Übergänge zu linken oder rechten Teilbäumen einen Pfad zum gewünschten Buchstaben sucht, erhält man die Morsecodierung des Buchstabens. *Materialien:* Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Binärbaum (Download Q2-I.1)

	implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), modifizieren Algorithmen und Programme (I), nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), interpretieren Fehlermeldungen und korrigieren den Quellcode (I), testen Programme systematisch anhand von Beispielen (I), stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D), stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).	
2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse `BinaryTree` (a) Analyse der Problemstellung, Ermittlung von Ob jekten, ihren Eigenschaften und Operationen im Anwendungskontext (b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Imple-mentationsdiagramms (c) Erarbeitung der Klasse `BinaryTree` und beispielhafte Anwendung der Operationen (d) Implementierung der Anwendung oder von Teilen der Anwendung (e) Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf		*Beispiel*: Informatikerbaum *als binärer Baum* In einem *binären Baum* werden die Namen und die Geburtsdaten von Informatikern lexikogra phisch geordnet abgespeichert. Alle Namen, die nach dieser Ordnung vor dem Namen im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem Namen im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.) Folgende Funktionalitäten werden benötigt: Einfügen der Informatiker-Daten in den Baum Suchen nach einem Informatiker über den Schlüssel Name Ausgabe des kompletten Datenbestands in nach Namen sortierter Reihenfolge *Materialien:* Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Binärbaum (Download Q2-I.2)

3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse BinarySearchTree (a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen (b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms, grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften (c) Erarbeitung der Klasse BinarySearchTree und Einführung des Interface Item zur Realisierung einer geeigneten Ordnungsrelation (d) Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums		*Beispiel*: *Informatiker*-baum als *Suchbaum* In einem binären Suchbaum werden die Namen und die Geburtsdaten von Informatikern lexikographisch geordnet abgespeichert. Alle Namen, die nach dieser Ordnung vor dem Namen im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem Namen im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.) Folgende Funktionalitäten werden benötigt: Einfügen der Informatiker-Daten in den Baum Suchen nach einem Informatiker über den Schlüssel Name Ausgabe des kompletten Datenbestands in nach Namen sortierter Reihenfolge *Materialien:* Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 Binärer Suchbaum (Download Q2-I.3)

4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen		*Beispiel:* Codierungsbäume (s.o.) oder Huffman-Codierung *oder* *Beispiel*: Buchindex Es soll eine Anwendung entwickelt werden, die anhand von Stichworten und zugehörigen Seitenzahlen ein Stichwortregister erstellt. Da die Stichwörter bei der Analyse des Buches häu- fig gesucht werden müssen, werden sie in der Klasse `Buchindex` als Suchbaum (Objekt der Klasse `BinarySearchTree`) verwaltet. Alle Inhalte, die nach einer Ordnung vor dem Inhalt im aktuellen Teilbaum ste- hen, sind in dessen linkem Teilbaum, alle die nach dem Inhalt im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.) *oder* *Beispiel*: Entscheidungsbäume (s.o.) *oder* *Beispiel*: Termbaum (s.o.) *oder* *Beispiel:* Ahnenbaum (s.o.)

Unterrichtsvorhaben Q2-II:

Thema: Endliche Automaten und formale Sprachen

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

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

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

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

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
1. Endliche Automaten (a) Vom Automaten in den Schülerinnen und Schülern bekannten Kontexten zur formalen Beschreibung eines endlichen Automaten (b) Untersuchung, Darstellung und Entwicklung endlicher Automaten	Die Schülerinnen und Schüler analysieren und erläutern die Eigenschaften endlicherAutomaten einschließlich ihres Verhaltens auf bestimmte Eingaben (A), analysieren und erläutern Grammatiken regulärer Sprachen (A), zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A), ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A), entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M), entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M), entwickeln zur akzeptierten Sprache eines Automaten die zugehörige Grammatik (M), entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M), modifizieren Grammatiken regulärer Sprachen (M), entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M), stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D), ermitteln die Sprache, die ein endlicher Automat akzeptiert (D). beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D).	*Beispiele:* Cola-Automat, Geldspielautomat, Roboter, Zustandsänderung eines Objekts „Auto“, Akzeptor für bestimmte Zahlen, Akzeptor für Teilwörter in längeren Zeichenketten, Akzeptor für Terme *Materialien:* Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.2 – Endliche Automaten, Formale Sprachen (Download Q2-II.1)

2. Untersuchung und Entwicklung von Grammatiken regulärer Sprachen (a) Erarbeitung der formalen Darstellung regulärer Grammatiken (b) Untersuchung, Modifikation und Entwicklung von Grammatiken (c) Entwicklung von endlichen Automaten zum Erkennen regulärer Sprachen die durch Grammatiken gegeben werden (d) Entwicklung regulä- rer		Beispiele: reguläre Grammatik für Wörter mit ungerader Parität, Grammatik für Wörter, die bestimmte Zahlen repräsentieren, Satzgliederungsgrammatik *Materialien: (s.o.)*
3. Grenzen endlicher Automaten		*Beispiele:* Klammerausdrücke, a^nb^n im Vergleich zu $(ab)^n$

Unterrichtsvorhaben Q2-III:

Thema: Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

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

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

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

Zeitbedarf: 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

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

Unterrichtsvorhaben Q2-IV:

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

2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

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

Überfachliche Grundsätze:

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

Fachliche Grundsätze:

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

2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

Auf der Grundlage von §13 - §16 der APO-GOSt sowie Kapitel 3 des Kernlehrplans Informatik für die gymnasiale Oberstufe hat die Fachkonferenz der Holzkamp-Gesamtschule 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 min
Grundkurse Q 1.1:	2 Klausuren je Halbjahr
Dauer der Klausuren:	135 min
Grundkurse Q 1.2:	2 Klausuren je Halbjahr
Dauer der Klausuren:	135 min
Grundkurse Q 2.1:	2 Klausuren
Dauer der Klausuren:	180 min
Grundkurse Q 2.2:	1 Klausur unter Abiturbedingungen

Anstelle der ersten 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.

Bewertungsraster zur Beurteilung von Klausuren
(Weitgehende Orientierung am Raster der Abiturprüfungen)

Note	Punkte	Prozent
sehr gut plus	15	95%-100%
sehr gut	14	90%-94,9%
sehr gut minus	13	85%-89,9%
gut plus	12	80%-84,9%
gut	11	75%-79,9%
gut minus	10	70%-74,9%
befriedigend plus	9	65%-69,9%
befriedigend	8	60%-64,9%
befriedigend minus	7	55%-59,9%
ausreichend plus	6	50%-54,9%
ausreichend	5	45%-49,9%
ausreichend minus	4	40%-44,9%
mangelhaft plus	3	33%-39,9%
mangelhaft	2	27%-32,9%
mangelhaft minus	1	20%-26,9%
ungenügend	0	0%-19,9%

2.3.2 Beurteilungsbereich Sonstige Mitarbeit

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

Leistungsaspekte

Mündliche Leistungen

- Qualität mündlicher Beiträge (bei neuen Themen ist nicht die Korrektheit sondern die richtige Einbringung von bereits vorher erlernten Elementen relevant)
- Quantität und Kontinuität mündlicher Beiträge
- Referate
- Vorstellung eigener Lernwege
- Kenntnis und Umgang mit Fachbegriffen
- Zusammenfassungen zur Vor- und Nachbereitung des Unterrichts
- Präsentation von Arbeitsergebnissen und Arbeitsprozessen (Projekte etc.)
- Mitarbeit in Partner-/Gruppenarbeitsphasen

Praktische Leistungen am Computer

- Implementierung, Test und Anwendung von Informatiksystemen
- Äußere Form von Programmtexten (Einrückung, Kommentierung, Beachtung von StyleGuides,…)

Sonstige schriftliche Leistungen

- Lernerfolgsüberprüfung durch kurze schriftliche Übungen (Notenvergabe siehe Raster zur Klausurbewertung)
- Qualität schriftlicher Beiträge, u.a. schriftliche Referate und Protokolle
- Quantität schriftlicher Beiträge

Angemessene Form und Inhalt der Heft- und Mappenführung
Bearbeitung von schriftlichen Aufgaben im Unterricht
Erstellung von Dokumentationen, Plakaten
Regelmäßigkeit, Umfang, termingerechte Einreichung und angemessene Form von Hausaufgaben
Lerntagebuch

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.

3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen

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

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

4 Qualitätssicherung und Evaluation

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