



ALFRED KRUPP SCHULE

Curriculum

des Faches

INFORMATIK

für die Sekundarstufen I & II

Stand April 2025

Alfred Krupp Schule

Margaretenstraße 40

45144 Essen

Tel.: (0201) 85 69 230

Fax: (0201) 85 69 231

E-Mail: info@daks-essen.de

Inhaltsverzeichnis

1. Die Fachschaft Informatik der Alfred Krupp Schule Essen	2
2. Unterrichtsvorhaben	3
2.1 Übersichtsraster Unterrichtsvorhaben Erprobungsstufe	4
2.2 Übersichtsraster Unterrichtsvorhaben Oberstufe	7
2.3 Konkretisierte Unterrichtsvorhaben	13
3. Grundsätze der fachmethodischen und fachdidaktischen Arbeit	47
4. Lehr und Lernmittel	48
5. Grundsätze der Leistungsbewertung und Leistungsrückmeldung	49
5.1 Beurteilungsbereich sonstige Mitarbeit:	49
5.2 Beurteilungsbereich Klausuren	51
5.3 Grundsätze der Leistungsrückmeldung und Beratung	52
6. Qualitätssicherung und Evaluation	52
7. Antidiskriminierung	53
8. Literaturverzeichnis	53

1. Die Fachschaft Informatik der Alfred Krupp Schule

Die Alfred Krupp Schule ist ein Stadtteilgymnasium in Essen Frohnhausen. Die Fachschaft Informatik besteht derzeit aus zwei Lehrkräften und verfügt über zwei Fachräume. Der große Informatikraum verfügt dabei über 16 DesktopPCs. Bei großen Klassen, d.h. in der Klasse 5/6 werden daher meist zusätzliche Laptops aus einem Laptopwagen mit 15 Laptops verwendet. Der kleinere Informatikraum mit 13 PCs wird vor allem für kleinere Kurse in der Oberstufe verwendet. Alle Arbeitsplätze sind an das schulinterne Rechnernetz angeschlossen, so dass Schülerinnen und Schüler über einen individuell gestaltbaren Zugang zum zentralen Server der Schule Zugriff auf ihre eigenen Daten haben, im Internet recherchieren oder schulische Aufgaben bearbeiten können. Zusätzlich verfügt die Fachschaft über 12 Lego-Spike-Roboter, die vor allem im Unterricht der Erprobungsstufe verwendet werden.

In Klasse 5 und 6 wird Informatik verpflichtend für alle Schülerinnen und Schüler unterrichtet. Da sich im Unterricht zeigte, dass in der Unterstufe eine Einzelstunde Informatik wenig hilfreich ist, insbesondere wenn es an Programmieraufgaben und die Arbeit mit dem PC geht, hat sich die Fachschaft dafür entschieden, vorzugsweise zweistündig zu unterrichten. Damit soll in Klasse 5 im zweiten Halbjahr und in Klasse 6 im ersten Halbjahr Informatik jeweils zweistündig unterrichtet werden.

In Klasse 9 und 10 gibt es jeweils einen Differenzierungskurs Informatik, der dreistündig, d.h. eine Doppelstunde und eine Einzelstunde unterrichtet wird.

In der Oberstufe wird Informatik an der AKS als Grundkurs angeboten und ist in der Regel gut besetzt, d.h. es gibt meistens zwei Kurse in der EF und einen oder zwei Grundkurse in der Q1 und Q2.

In der Sekundarstufe II bieten wir Informatik unabhängig von der Wahl im Differenzierungsbereich als dreistündigen Grundkurs an. Es ist möglich, das Fach Informatik im Abitur als 4. Abiturfach (mündlich) oder als 3. Abiturfach (schriftlich) zu belegen.

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

Der Unterricht der Sekundarstufe II wird mit Hilfe der Programmiersprache Java in der Entwicklungsumgebung BlueJ durchgeführt. In der Einführungsphase kommt dabei zusätzlich die Mikroumgebung Greenfoot¹ zum Einsatz, welche den Einstieg in die Programmierung erleichtert.

Durch projektorientiertes 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.

Das hier vorliegende schulinterne Curriculum setzt den Kernlehrplan², sowie die Vorgaben für das Zentralabitur³ um, die vor allem den wissenschaftlichen Aspekt der Informatik in den Blick nehmen. Dazu gehören die Einführung in eine objektorientierte Programmiersprache (Java), das objektorientierte Modellieren in UML und Teilbereiche der theoretischen Informatik.

2. 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. Im Folgenden wird zuerst das Curriculum der Klassen 5 und 6 und dann das der Oberstufe dargestellt.

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

Im „Übersichtsraster Unterrichtsvorhaben“ (Kapitel 2) 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 „konkreter Unterrichtsvorhaben“ (Kapitel 2.2) 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 3 bis 5 zu entnehmen sind.

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

² (Lehrplan SII, 2014)

³ (Informatik, 2017)

2.1 Übersichtsraster Unterrichtsvorhaben Erprobungsstufe

Klasse 5				
Inhalte/ Inhaltsfelder	Kompetenzen Die Schülerinnen und Schüler sollen:	Fachlicher Kontext/ Prozessbezogene Kompetenzen	Inhaltliche Konkretisierung	Vorschläge für Fächer verbindendes Arbeiten
Informatiksysteme und Informatik, Mensch und Gesellschaft	- Benennen Beispiele für Informatiksysteme aus ihrer Erfahrungswelt (DI)⁴ (MKR 1.1) - Benennen Grundkomponenten und beschreiben die Funktionen (DI) - Beschreiben das EVA-Prinzip (DI) - Beschreiben die Bedeutung von Informatiksystemen in der Lebens- und Arbeitswelt (KK) - Beschreiben anhand von Beispielen Verarbeitung und Nutzung personenbezogener Daten (DI) (MKR 1.4) - Vergleichen Datenträger hinsichtlich Ort, Kapazität, Datensicherheit (A)	- Begründen die Wahl eines Informatiksystems für best. Ansprüche (A) (MKR 1.1) - Wenden Prinzipien der strukturierten Dateiverwaltung an (DI) (MKR 1.3) - Setzen zielgerichtet Informatiksysteme zur Verarbeitung von Daten ein (MI) - Setzen Informatiksysteme zur Kommunikation und Kooperation ein.(KK) (MKR 3.1) - Schreiben und Versenden einer eMail mit Adresse und Kopfzeile (Methodencurriculum) (MKR 3.3) - Beschreiben anhand von ausgewählten Beispielen(z.B. angehängten Dateien und Emailadressen die Verarbeitung und	- Vergleichen von Handys, Tablets, Laptops und Desktop-PCs - Innenleben eines PCs untersuchen - Dateiträger und Speichermöglichkeiten kennenlernen - Benutzernamen, Accounts und Kennwörter - Kennwortsicherheit - Wer sieht meine Daten?	SoWi in der Sek.I und Politik

⁴A: Argumentieren DI: Darstellen und Interpretieren, MI: Modellieren und Implementieren, KK: Kommunizieren und Kooperieren

	• Erkennen von PhishingMails und kennen Gefahren dabei • Kennen Merkmale sicherer Internetseiten	• Nutzung personenbezogener Daten (DI) • Beschreiben und Nutzen Maßnahmen zum Schutz personenbezogener Daten (auch im Hinblick auf Speicherorte) (A)		
Informationen und Daten	• Erläutern den Datenbegriff anhand von Beispielen aus ihrer Erfahrungswelt (A) • Erläutern den Zusammenhang und die Bedeutung von Informationen und Daten (A)	• Codieren Daten und Informationen mit Hilfe des Binärsystems (MI) • Interpretieren Daten als Informationen in einem Kontext (DI) (MKR 6.1) • Verwenden (und erläutern) ein einfaches Transpositionsverfahren (DI und MI) • Vergleichen verschiedene Verschlüsselungsverfahren anhand von ausgewählten Sicherheitsaspekten (DI)	• Datengrößen • Binärsystem • ASCII • Cäsar • Transpositionsverfahren • Steganographie	Mathe – Zahlensysteme
Klasse 6				
Inhalte/ Inhaltsfelder	Kompetenzen • Die Schülerinnen und Schüler sollen:	Fachlicher Kontext/ • Prozessbezogene Kompetenzen	Inhaltliche Konkretisierung	Vorschläge für Fächer verbindendes Arbeiten

Algorithmen und Programmieren	• Formulieren zu Abläufen aus dem Alltag eindeutige Handlungsvorschriften (DI) (MKR 6.2) • Überführen Handlungsvorschriften in einen Programmablaufplan (MI) (MKR 6.3) • Identifizieren in Handlungsvorschriften Anweisungen und die algorithmischen Grundstrukturen Sequenz, Verzweigung und Schleife (MI) • Implementieren Algorithmen in einer visuellen Programmiersprache (MI) (MKR 6.3) • Identifizieren in Handlungsvorschriften Anweisungen und die algorithmischen Befehle (MKR 6.2)	• Formulieren zu Abläufen aus dem Alltag eindeutige Handlungsvorschriften (DI) • erstellen informatische Modelle zu gegebenen Sachverhalten (MI) • beschreiben einfache Darstellungen von informatischen Sachverhalten (DI) • verwenden Algorithmen um Probleme zu lösen (MI)	• Nutzen der Lego-Spikeroboter und Scratch • Anweisungen (Definition + Eigenschaften) • Sequenzen als Aneinanderreihung von Anweisungen • Funktionen als Pakete von Sequenzen • Bedingte Anweisungen und Verzweigungen • Schleifen	Mathematik – Algorithmisches Zeichnen von Vielecken Musik – Durch einen Algorithmus festgelegtes Ausgeben von Tönen, die zu einer individuellen Melodie resultieren Kunst. Erstellung von digitalen künstlerische Produkte
Automaten und künstliche Intelligenz	• Erläutern die Funktionsweise eines Automaten aus ihrer Lebenswelt (A) (MKR 6.2) • Benennen Anwendungsbeispiele künstlicher Intelligenz aus ihrer Lebenswelt (A) (MKR 6.2) • Beschreiben die Grundlegende Funktion neuronaler Netze (KK)	• benennen Anwendungsbeispiele künstlicher Intelligenz aus ihrer Lebenswelt (A) • Stellen Abläufe am Automaten grafisch dar (DI) • Stellen das Prinzip eines Entscheidungsbaumes als ein Prinzip des maschinellen Lernens dar (DI) • Beurteilen die Anwendung von KI in verschiedenen Sachzusammenhängen (A)	• KI im Alltag • Chancen, Grenzen und Risiken der KI • Entscheidungsbaum und optimieren • Lernen durch Training (Wie werden KIs trainiert?)	SoWi in der Sek.I und Politik

2.2 Übersichtsraster Unterrichtsvorhaben Oberstufe

Einführungsphase

Thema / Unterrichtsvorhaben	Zentrale Kompetenzen	Inhaltsfelder	Inhaltliche Schwerpunkte
Unterrichtsvorhaben EF-I			
Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten	• Argumentieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren • MKR 1.1 – 1.4	• Informatiksysteme • Informatik, Mensch und Gesellschaft	• Einzelrechner • Dateisystem • Internet • Einsatz von Informatiksystemen
Unterrichtsvorhaben EF-II			
Grundlagen der objektorientierten Analyse, Modellierung, Implementierung	• Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren • MKR 6.1 bis 6.3	• Daten und ihre Strukturierung • Formale Sprachen und Automaten	• Objekte und Klassen • Syntax und Semantik einer Programmiersprache
Unterrichtsvorhaben EF-III			
Algorithmische Grundstrukturen in Java	• Argumentieren • Modellieren • Implementieren • Kommunizieren und Kooperieren • MKR 6.1 – 6.3	• Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten	• Objekte und Klassen • Syntax und Semantik einer Programmiersprache • Analyse, Entwurf und Implementierung einfacher Algorithmen

Unterrichtsvorhaben EF-IV			
Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand lebensnaher Anforderungsbeispiele	• Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren	• Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten	• Objekte und Klassen • Syntax und Semantik einer Programmiersprache • Analyse, Entwurf und Implementierung einfacher Algorithmen
Unterrichtsvorhaben EF-V			
Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes	• Argumentieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren • MKR 1.4 und 4.4	• Informatik, Mensch und Gesellschaft • Informatiksysteme	• Wirkungen der Automatisierung • Geschichte der automatischen Datenverarbeitung • Digitalisierung • Internet
Summe EF: ca. 60 Stunden			

Qualifikationsphase 1

Thema / Unterrichtsvorhaben	Zentrale Kompetenzen	Inhaltsfelder	Inhaltliche Schwerpunkte
Unterrichtsvorhaben Q1-I			
Endliche Automaten und formale Sprachen	• Argumentieren • Modellieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren	• Endliche Automaten und formale Sprachen	• Endliche Automaten • Grammatiken regulärer Sprachen • Möglichkeiten und Grenzen von Automaten und formalen Sprachen
Unterrichtsvorhaben Q1-II			
Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung	• Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren • MKR 6.3	• Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten • Informatiksysteme	• Objekte und Klassen • Analyse, Entwurf und Implementierung von Algorithmen • Syntax und Semantik einer Programmiersprache • Nutzung von Informatiksystemen
Unterrichtsvorhaben Q1-III			

Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen	• Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren	• Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten	• Objekte und Klassen • Analyse, Entwurf und Implementierung von Algorithmen • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache
Unterrichtsvorhaben Q1-IV			
Suchen und Sortieren	• Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren	• Algorithmen • Formale Sprachen und Automaten	• Analyse, Entwurf und Implementierung von Algorithmen • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache
Unterrichtsvorhaben Q1-V			
Sicherheit und Datenschutz in Netzstrukturen	• Argumentieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren • MKR 2.4, 3.4 und 4.4	• Informatiksysteme • Informatik, Mensch und Gesellschaft	• Einzelrechner und Rechnernetzwerke • Sicherheit • Nutzung von Informatiksystemen • Wirkungen der Automatisierung • Grenzen der Automatisierung
Summe Q1: 80 Stunden			

Qualifikationsphase 2

Thema / Unterrichtsvorhaben	Zentrale Kompetenzen	Inhaltsfelder	Inhaltliche Schwerpunkte
Unterrichtsvorhaben Q2-I 24 Stunden			
Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen	• Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren	• Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten	• Objekte und Klassen • Analyse, Entwurf und Implementierung von Algorithmen • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache
Unterrichtsvorhaben Q2-II			
Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten	• Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren	• Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten • Informatik, Mensch und Gesellschaft	• Datenbanken • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache • Sicherheit
Unterrichtsvorhaben Q2-III			
Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit	• Argumentieren • Kommunizieren und Kooperieren • MKR 6.4	• Informatiksysteme • Informatik, Mensch und Gesellschaft	• Einzelrechner und Rechnernetzwerke • Grenzen der Automatisierung

Unterrichtsvorhaben Q2-IV			
Wiederholung der Inhalte aus Q1 und Q2, Prüfungsvorbereitung	• Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren	• alle oben genannten	• alle oben genannten
Summe Q2: ca. 60 Stunden			

2.3 Konkretisierte Unterrichtsvorhaben

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

In der Einführungsphase wird zum Einstieg in die Programmierung Greenfoot⁵ verwendet. Im weiteren Verlauf wird BlueJ⁶ verwendet.

In der Qualifikationsphase werden die Unterrichtsvorhaben unter Berücksichtigung der Vorgaben für das Zentralabitur Informatik in NRW konkretisiert⁷.

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).

⁵ (Michael Kölling, 2025)

⁶ (Oracle, 2025)

⁷ (Informatik, 2017)

Unterrichtsvorhaben EF-I

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

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

Vorhabenbezogene Konkretisierung:

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

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

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

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

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
• Information, deren Kodierung und Speicherung (a) Informatik als Wissenschaft der Verarbeitung von Informationen (b) Darstellung von Informationen in Schrift, Bild und Ton	Die Schülerinnen und Schüler • beschreiben und erläutern den Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A),	<i>Beispiel:</i> Textkodierung Kodierung und Dekodierung von Texten mit unbekannten Zeichensätzen (z.B. Wingdings) <i>Beispiel:</i> Bildkodierung

(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.)	• nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D), • interpretieren Binärcodes als Zahlen und Zeichen (D),	Kodierung von Bildinformationen in Raster- und Vektorgrafiken
• Informations- und Datenübermittlung in Netzen (a) „Sender-Empfänger-Modell“ und seine Bedeutung für die Eindeutigkeit von Kommunikation (b) Informatische Kommunikation in Rechnernetzen (c) Grundlagen der technischen Umsetzung von Rechnerkommunikation (d) Richtlinien zum verantwortungsvollen Umgang in Rechnernetzen	• nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K).	<i>Beispiel:</i> Schulnetzwerk (z.B. Benutzeranmeldung, Netzwerkordner, Zugriffsrechte, Client-Server) <i>Beispiel:</i> Internet (z.B. Netzwerkadresse, Paketvermittlung, Protokoll)
• Aufbau informatischer Systeme (a) Identifikation typischer Komponenten informatischer Systeme und anschließende Beschränkung auf das Wesentliche, Herleitung der „Von-Neumann-Architektur“ (b) Identifikation des EVA-Prinzips (Eingabe-Verarbeitung-Ausgabe) als Prinzip der Verarbeitung von Daten und Grundlage der „Von-Neumann-Architektur“		<i>Material:</i> Demonstrationshardware Durch Demontage eines Demonstrationsrechners entdecken Schülerinnen und Schüler die verschiedenen Hardwarekomponenten eines Informatiksystems, sowie deren Komposition.

Unterrichtsvorhaben EF-II

Thema: Grundlagen der objektorientierten Analyse, Modellierung, Implementierung

Leitfrage: Wie lassen sich Gegenstandsbereiche informatisch modellieren und innerhalb einer Entwicklungsumgebung 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, sodass ein Verständnis von den Begriffen Klasse und Objekt geschaffen wird.

Im Anschluss lernen die Schülerinnen und Schüler die Greenfoot Umgebung im Rover-Szenario⁸ kennen, platzieren Objekte und interagieren mit den enthaltenen Attributen und Methoden.

Da bei der Umsetzung der ersten Programme konsequent auf die Verwendung von Kontrollstrukturen verzichtet wird und da 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. Die Arbeit an den Aufgaben wird unmittelbar zum nächsten Unterrichtsvorhaben führen. Dort stehen unter anderem Kontrollstrukturen im Mittelpunkt.

⁸ (NGB, 2020)

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 z.B. an der Tafel visualisiert und mit sinnvollen Attributen und Methoden versehen. (c) Manche Objekte sind prinzipiell typgleich und werden so zu einer Objektsorte bzw. Klasse 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).	<i>Beispiele:</i> Klasse Auto, Klasse Haus
2. Erste Schritte mit Greenfoot (a) Klasse und Objekt in der Entwicklungsumgebung kennen lernen (b) Objekte erstellen und Funktionsweise der gegebenen Methoden erfassen (c) Methoden mit und ohne Rückgabewert unterscheiden (d) Werte von Attributen in Objekten inspizieren		<i>Material:</i> Buch, Greenfoot
3. Quelltexte schreiben (a) Begriffsbildung „kompilieren“, „Maschinencode“, „Quelltext“ und Lifecycle eines Java-Programms (b) Erstellen eines einfachen, linearen Algorithmus (c) Strukturierung von Quelltext mit Hilfe von Methoden		

Unterrichtsvorhaben EF-III

Thema: Algorithmische Grundstrukturen in Java

Leitfragen: Wie lassen sich Aktionen von Objekten flexibel realisieren?

Vorhabenbezogene Konkretisierung:

Der Schwerpunkt dieses Unterrichtsvorhabens liegt auf der Erarbeitung von Kontrollstrukturen und den daraus resultierenden Algorithmen, die den Rover durch seine (sich verändernde) Welt führen. Die Erstellung der Algorithmen findet zunächst in Flussdiagrammen, später auch als Quelltext statt. In den zu lösenden Szenarien sind zum Teil Objekte vorgegeben, die in die Lösung einbezogen werden sollen. Für die Umsetzung werden Kontrollstrukturen in Form von Schleifen und bedingten Anweisungen benötigt und eingeführt. Später werden diese mit Variablen ergänzt. Da die zu entwickelnden Algorithmen zunehmend umfangreicher werden, werden systematische Vorgehensweisen zur Entwicklung von Algorithmen thematisiert

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Algorithmen darstellen / bedingte Anweisungen (a) Aufbau und Komponenten eines Flussdiagramms zur Darstellung eines Algorithmus (b) Verwendung von Methoden zur Zusammenfassung mehrerer Befehle (c) Bedingte Anweisungen mit einer Bedingung und else-Block (d) Analyse eines Quelltextes mit bedingten Anweisungen (e) Verschachtelte Bedingungen	Die Schülerinnen und Schüler • 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), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen zu (M),	Buch, Greenfoot

(f) Verknüpfung mehrerer Bedingungen mit den logischen Operatoren &&, und !	• modifizieren einfache Algorithmen und Programme (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), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I).	
2. Wiederholungen (a) Mehrfaches Wiederholen von Quelltext (b) Analyse mehrerer Schleifen (c) sinnvolles Anwenden in eigenen Programmen		
3. Variablen (a) Einführung über eine Zählvariable (b) Konstanten (c) Elementare Datentypen kennen lernen (d) Vergleichsoperatoren (e) Arithmetische Operatoren (f) Anwendung in eigenen Programmen		
4. Anwendungsbeispiel (a) Erstellung eines Spiels für den Rover (b) Reaktion auf Benutzereingaben (c) Einzelne Methoden mit bestimmten Aufgaben programmieren (d) Zahlreiche Möglichkeiten zur Differenzierung, wie Highscoreliste, Einfügen eigener Bilder		

Unterrichtsvorhaben EF-IV

Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand lebensnaher Anforderungsbeispiele

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

Vorhabenbezogene Konkretisierung:

In Abgrenzung zu dem vorherigen Unterrichtsvorhaben liegt hier der Akzent auf der Entwicklung von Objekt -und Klassenbeziehungen. In diesem Unterrichtsvorhaben erfolgt der Umstieg von Greenfoot mit Rover auf BlueJ, um die Schülerinnen und Schüler an die für sie gängige Entwicklungsumgebung zu gewöhnen. Es werden, ausgehend von der Realität, über Objektidentifizierung und Entwurf bis hin zur Implementation kleine Softwareprodukte in Teilen oder ganzheitlich erstellt oder dekonstruiert bzw. modifiziert.

Zuerst identifizieren die Schülerinnen und Schüler Objekte und stellen diese dar. Aus diesen Objekten werden Klassen und ihre Beziehungen in Entwurfs- und später Implementationsdiagrammen visualisiert. In diesem Kontext wird auch die Vererbung von Klassen und das Geheimnisprinzip erarbeitet.

Für die Umsetzung eines etwas größeren Projektes werden das Wasserfallmodell, die Erstellung einer graphischen Benutzeroberfläche, sowie Arrays eingeführt.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Umsetzung von Anforderungen in Entwurfsdiagramme (a) Aus Anforderungsbeschreibungen werden Objekte mit ihren Eigenschaften identifiziert (b) Gleichartige Objekte werden in Klassen (Entwurf) zusammengefasst und um Datentypen und Methoden erweitert	Die Schülerinnen und Schüler • analysieren und erläutern eine objektorientierte Modellierung (A), • stellen die Kommunikation zwischen Objekten grafisch dar (M), • ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),	<i>Beispiel:</i> Klasse Scooter, Klasse AppVererbung z.B. mit Klasse „Fahrzeuge“.

- modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M),
- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M),
- ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),
- modellieren Klassen unter Verwendung von Vererbung (M),
- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),
- testen Programme schrittweise anhand von Beispielen (I),
- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),
- analysieren und erläutern einfache Algorithmen und Programme (A)
- modifizieren einfache Algorithmen und Programme (I),
- entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M).
- stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D),
- dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D).

2. Implementationsdiagramme (a) Erweiterung des Entwurfsdiagramms um Konstruktoren und get- und set-Methoden / Geheimnisprinzip (b) Festlegung von Datentypen in Java, sowie von Rückgaben und Parametern (c) Entwicklung von Klassendokumentationen (d) Vererbungsbeziehungen und Vorteile daraus (e) Implementation der Diagramme, kleinere Aufgaben zur eigenständigen Planung und Implementation		
3. Erweiterung durch GUI und Arrays (a) Nutzung dokumentierter Klassenbibliotheken zur Implementation einer grafischen Benutzeroberfläche (b) Konzept von Action-Listenern (c) Implementation kleinerer Programme mit GUI (d) Verwendung und Nutzen von Arrays (e) Implementation von Aufgaben mit Arrays		<i>Beispiel:</i> Erstellung einer GUI für die Scooter-App

4. Softwareprojekt (a) Wasserfallmodell (b) Durchlaufen der Schritte des Wasserfallmodells zur Dokumentation und Programmierung eines Softwareprodukts		<i>Beispiel:</i> Tic-Tac-Toe, ein wissenschaftlicher Taschenrechner, „Wer wird Millionär“-Klon
--	--	---

Unterrichtsvorhaben EF-V

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 Präsentationen 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 Datenschutzgesetzes.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Selbstständige Erarbeitung von Themen durch die Schülerinnen und Schüler (a) Mögliche Themen zur Erarbeitung in Kleingruppen: • „Eine kleine Geschichte der Digitalisierung: vom Morsen zum modernen Digitalcomputer“	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),	<i>Beispiel:</i> 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.

• „Eine kleine Geschichte der Kryptographie: von Caesar zur Enigma“ • „Von Nullen, Einsen und mehr: Stellenwertsysteme und wie man mit ihnen rechnet“ • „Big Data im Hinblick auf Werbeindustrie und Geheimdienste“ • „Auswirkungen der Digitalisierung: Veränderungen der Arbeitswelt und Datenschutz“ (b) Vorstellung und Diskussion durch Schülerinnen und Schüler	• interpretieren Binärcodes als Zahlen und Zeichen (D), • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation. (K).	<i>Materialien:</i> Schülerinnen und Schüler recherchieren selbstständig im Internet, in der Schulbibliothek, in öffentlichen Bibliotheken, usw.
2. Vertiefung des Themas Datenschutz (a) Erarbeitung grundlegender Begriffe des Datenschutzes (b) Problematisierung und Anknüpfung an die Lebenswelt der Schülerinnen und Schüler (c) Diskussion und Bewertung von Fallbeispielen aus dem Themenbereich „Datenschutz“		<i>Beispiel:</i> Aktuelle Medienberichterstattung. <i>Materialien:</i> Materialblatt zum Bundesdatenschutzgesetz⁹

⁹ (Qualis, 2017)

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: 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. Dies wird 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, realisiert.

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

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Umsetzung von Anforderungen in Entwurfsdiagramme (a) Vom den Schülerinnen und Schülern bekannten, realen Automaten zur formalen Beschreibung eines endlichen Automaten (b) Untersuchung, Darstellung und Entwicklung endlicher Automaten	Die Schülerinnen und Schüler • analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens auf bestimmte Eingaben (A), • analysieren und erläutern Grammatiken regulärer Sprachen (A), • zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A), • ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A), • entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M), • entwickeln 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	<i>Beispiele:</i> <i>Cola-Automat, Geldspielautomat, Roboter, Zustandsänderung eines Objekts „Auto“, Akzeptor für bestimmte Zahlen, Akzeptor für Teilwörter in längeren Zeichenketten, Akzeptor für Terme</i> <i>Materialien:</i> <i>Endliche Automaten, Formale Sprachen¹⁰</i>
2. Untersuchung und Entwicklung von Grammatiken regulärer Sprachen (a) Erarbeitung der formalen Darstellung regulärer Grammatiken (b) Untersuchung, Modifikation und Entwicklung von Grammatiken (c) Entwicklung von endlichen Automaten zum Erkennen regulärer Sprachen die durch Grammatiken gegeben werden (d) Entwicklung regulärer Grammatiken zu endlichen Automaten	(siehe oben)	<i>Beispiele:</i> reguläre Grammatik für Wörter mit ungerader Parität, Grammatik für Wörter, die bestimmte Zahlen repräsentieren, Satzgliederungsgrammatik Materialien: (s.o.)

¹⁰ (Qualis, 2020)

3. Grenzen endlicher Automaten	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: Klammerausdrücke, $a^n b^n$ im Vergleich zu $(ab)^n$
--	--	---

Unterrichtsvorhaben Q1-II

Thema: Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung

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

Vorhabenbezogenen Konkretisierung:

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

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

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Wiederholung und Erweiterung der objektorientierten Modellierung und Programmierung durch Analyse und Erweiterung eines kontextbezogenen Beispiels (a) Analyse der Problemstellung (b) Analyse der Modellierung (Implementationsdiagramm) (c) Erweiterung der Modellierung im Implementationsdiagramm (Vererbung, abstrakte Klasse) (d) Kommunikation zwischen mindestens zwei Objekten (grafische Darstellung) (e) Dokumentation von Klassen (f) Implementierung der Anwendung oder von Teilen der Anwendung	Die Schülerinnen und Schüler • analysieren und erläutern objektorientierte Modellierungen (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), • ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen (D), • stellen die Kommunikation zwischen Objekten grafisch dar (D)	

Unterrichtsvorhaben Q1-III

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 erläutert. Anschließend werden für die Anwendung notwendige Klassen modelliert und implementiert. Eine Klasse für eine, den Anforderungen der Anwendung entsprechende, Oberfläche sowie die Klasse Stack 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 First-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.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Die Datenstruktur Stack 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	Die Schülerinnen und Schüler • erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), • analysieren und erläutern Algorithmen und Programme (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),	<i>Beispiel:</i> Heftstapel In einem Heftstapel soll das Heft einer Schülerin gefunden werden. <i>Beispiel:</i> Kisten stapeln In einem Stapel nummerierter Kisten soll eine bestimmte Kiste gefunden und an einen Kunden geliefert werden. Dazu müssen Kisten auf

	• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nicht-lineare Datensammlungen zu (M), • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • modifizieren Algorithmen und Programme (I), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen (I),	verschiedene Stapel gestapelt und wieder zurückgestellt werden.
2. 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		<i>Beispiel:</i> Patientenwarteschlange (jeder kennt seinen Nachfolger bzw. alternativ: seinen Vorgänger) Sobald ein Patient in einer Arztpraxis eintrifft, werden sein Name und seine Krankenkasse erfasst. Die Verwaltung der Patientenwarteschlange geschieht über eine Klasse, die hier als Wartezimmer bezeichnet wird. Patienten können sich zusätzlich in die Warteschlange zum Blutdruckmessen einreihen. Objekte werden von zwei Schlangen verwaltet.
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	stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D).	<i>Beispiel:</i> Abfahrtslauf Bei einem Abfahrtslauf kommen die Skifahrer nacheinander an und werden nach ihrer Zeit in eine Rangliste eingeordnet. Diese Rangliste wird in einer Anzeige ausgegeben. Ankommende Abfahrer müssen an jeder Stelle der Struktur, nicht nur am Ende oder Anfang eingefügt werden können.
4. Vertiefung - Anwendungen von Listen Stapeln oder Schlangen in mindestens einem weiteren Kontext		<i>Beispiel:</i> Aufgaben aus dem Zentralabitur aus den letzten Jahren

Unterrichtsvorhaben Q1-IV

Thema: Suchen und Sortieren

Leitfragen: *Wie können Objekte bzw. Daten effizient sortiert werden, so dass eine schnelle Suche möglich wird? Wie kann man gespeicherte Informationen effizient (wieder-)finden?*

Vorhabenbezogene Konkretisierung:

Dieses Unterrichtsvorhaben beschäftigt sich mit der Erarbeitung von Such- und Sortialgorithmen.

In einem Anwendungskontext werden zunächst Informationen in einer linearen Liste bzw. einem Array 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 sollen verschiedene Sortialgorithmen kennen gelernt und hinsichtlich Speicherplatzbedarf, Komplexität, Stabilität usw. untersucht und in einem Beispielprojekt implementiert werden.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Suchen von Daten in Listen und Arrays (a) Lineare Suche in Listen und in Arrays (b) Binäre Suche in Arrays als Beispiel für rekursives Problemlösen (c) Untersuchung der beiden Suchverfahren hinsichtlich ihrer Effizienz (Laufzeitverhalten, Speicherbedarf)	Die Schülerinnen und Schüler • analysieren und erläutern Algorithmen und Programme (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),	<i>Beispiele:</i> Abfahrtslauf aus Unterrichtsvorhaben Q1-III, Suche nach einer Adresse in einer Adressverwaltung

2. Sortieren in Listen und Arrays - Entwicklung und Implementierung von iterativen und rekursiven Sortierverfahren (a) Analyse eines Sortieralgorithmus hinsichtlich kennen gelernter Merkmale (b) Implementation des Algorithmus in ein Beispielprojekt (c) Präsentation der Ergebnisse	• beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M), • modifizieren Algorithmen und Programme (I), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen (I), stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).	<i>Beispiel:</i>In Form von Referaten mit kleinerer Ausarbeitung können die Sortieralgorithmen Bubblesort, Insertionsort, Selectionsort, Gnomesort, Quicksort und Mergesort untersucht und für Arrays und/oder Listen in ein Beispielprojekt implementiert werden
3. Untersuchung der Effizienz der vorgestellten Sortierverfahren im Vergleich (a) Grafische Veranschaulichung der Sortierverfahren (b) Anzahl der Vergleichsoperationen ermitteln (c) Beurteilung der Effizienz der Sortierverfahren, auch hinsichtlich worst-average- und best-case	(siehe oben)	Beispiel: Es können hier exemplarisch zwei Verfahren ausgewählt 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? Wie können sensible Daten geschützt werden?

Vorhabenbezogene Konkretisierung:

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

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	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	<i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben, Verschlüsselung Q1.5 - Zugriff auf Daten in Netzwerken ¹¹ Erarbeitung der Verschiedenen kryptografischen Verfahren und Methoden zum Schutz von sensiblen Daten kann über Referate erfolgen

¹¹<http://www.standardsicherung.schulministerium.nrw.de/materialdatenbank/nutzersicht/getFile.php?id=6320>

kryptografische Verfahren (Cäsar-, Vigenère-, RSA-Verfahren) als Methoden Daten im Netz verschlüsselt zu übertragen (d) Wie schütze ich mich vor dem Missbrauch meiner Daten?	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).	
2. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht		Materialien: Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1 5 - Datenschutz beim Videocenter¹²

¹²<http://www.standardsicherung.schulministerium.nrw.de/materialdatenbank/nutzersicht/getFile.php?id=6321>

Unterrichtsvorhaben Q2-I

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

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

Vorhabenbezogene Konkretisierung:

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

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

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

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

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

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

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 anhand 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 syntaktische Korrektheit und die Funktionalität von Programmen (A), • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nicht-lineare Datensammlungen zu (M), • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M),	<i>Beispiel:</i> Termbaum Der Aufbau von Termen wird mit Hilfe von binären Baumstrukturen verdeutlicht. oder <i>Beispiel:</i> Ahnenbaum Die binäre Baumstruktur ergibt sich daraus, dass jede Person genau einen Vater und eine Mutter hat.
2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse BinaryTree (a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext (b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms (c) Erarbeitung der Klasse BinaryTree und beispielhafte Anwendung der Operationen (d) Implementierung der Anwendung oder von Teilen der Anwendung	(siehe oben)	<i>Beispiel:</i> Informatikerbaum als binärer Baum In einem <i>binären Baum</i> werden die Namen und die Geburtsdaten von Informatikern lexikographisch geordnet abgespeichert. Alle Namen, die nach dieser Ordnung vor dem Namen im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem Namen im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.) Folgende Funktionalitäten werden benötigt: Einfügen der Informatiker-Daten in den Baum, Suchen nach einem Informatiker über den Schlüssel Name,

(e) Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf	• 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).	Ausgabe des kompletten Datenbestands in nach Namen sortierter Reihenfolge
3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Nutzung der Klasse BinarySearchTree (a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen (b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms, (c) grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften (d) Erarbeitung der Klasse BinarySearchTree und Einführung des Interface Item zur Realisierung einer geeigneten Ordnungsrelation (e) Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums		Beispiel: Informatikerbaum 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
4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen		<i>Beispiel:</i> Aufgaben aus dem Zentralabitur aus den letzten Jahren

Die in den Beispielen genannten Materialien finden sich unter: <http://www.standardsicherung.schulministerium.nrw.de/materialdatenbank/nutzersicht/materialeintrag.php?matId=4354>

Unterrichtsvorhaben Q2-II

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.

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 Datenbankbeispiel	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 Datenbankmodellierung (M), • modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M), • bestimmen Primär- und Sekundärschlüssel (M), • überführen Datenbankschemata in vorgegebene Normalformen (M),	

2. Modellierung von relationalen Datenbanken (a) Entity-Relationship-Diagramm • Ermittlung von Entitäten, zugehörigen Attributen, Relationen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms • Erläuterung und Modifizierung einer Datenbankmodellierung (b) Entwicklung einer Datenbank aus einem Datenbankentwurf • Modellierung eines relationalen Datenbankschemas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln (c) Redundanz, Konsistenz und Normalformen • Untersuchung einer Datenbank hinsichtlich Konsistenz und Redundanz in einer Anwendungssituation • Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten)	• verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I), • ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D), • stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D), • überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D).	<i>Beispiel: Reederei</i> 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. <i>Beispiel: Schulverwaltung</i> In einer Software werden die Schulhalbjahre, Jahrgangsstufen, Kurse, Klassen, Schüler, Lehrer und Noten einer Schule verwaltet. Man kann dann ablesen, dass z.B. Schüler X von Lehrer Y im 2. Halbjahr des Schuljahrs 2011/2012 in der Jahrgangsstufe 9 im Differenzierungsbereich im Fach Informatik die Note „sehr gut“ erhalten hat. Dazu ist die Datenbank zu modellieren, ggf. zu normalisieren und im Datenbanksystem umzusetzen. Weiter sollen sinnvolle Abfragen entwickelt werden und das Thema Datenschutz besprochen werden.
--	--	---

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.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, 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	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).	<i>Beispiel:</i> Addition von 4 zu einer eingegeben Zahl mit einem Rechnermodell <i>Materialien:</i>

(c) Analyse und Erläuterung der Funktions-weise eines einfachen maschinennahen Programms		Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 –Von-Neumann-Architektur und maschinennahe Programmierung ¹³
2. Grenzen der Automatisierbarkeit (a) Vorstellung des Halteproblems (b) Unlösbarkeit des Halteproblems (c) Beurteilung des Einsatzes von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen		<i>Beispiel:</i> Halteproblem <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator ¹⁴

¹³<http://www.standardsicherung.schulministerium.nrw.de/materialdatenbank/nutzersicht/getFile.php?id=6329>

¹⁴<http://www.standardsicherung.schulministerium.nrw.de/materialdatenbank/nutzersicht/getFile.php?id=6330>

Unterrichtsvorhaben Q2-IV

Thema: Wiederholung der Inhalte aus Q1 und Q2, Prüfungsvorbereitung

Leitfragen: Welche Inhalte sollen ausführlicher wiederholt werden? Was ist besonders zu beachten im Zentralabitur bzw. in der mündlichen Prüfung?

Vorhabenbezogene Konkretisierung:

Verschiedene Inhalte aus der Oberstufe sollen noch einmal thematisiert werden. Die Schülerinnen und Schüler erhalten die Möglichkeit, sich mit den im Unterricht behandelten Themen noch einmal auseinanderzusetzen. Dies kann z.B. in Form von Ausarbeitungen geschehen. Parallel dazu können mündliche Prüfungssimulationen durchgeführt werden.

In diesem Unterrichtsvorhaben soll ganz auf die individuellen Wünsche der Schülerinnen und Schüler eingegangen werden. Daher gibt es keine festen Unterrichtssequenzen. Es können sämtliche in den vorhergehenden Unterrichtsvorhaben angesprochene Kompetenzen entwickelt bzw. vertieft werden.

3. Grundsätze der fachmethodischen und -didaktischen Arbeit

Unter Berücksichtigung des Schulprogramms hat die Fachkonferenz Informatik der Alfred Krupp Schule die folgenden fachmethodischen und fachdidaktischen Grundsätze beschlossen. Die AKS legt Wert auf individuelles Lernen und Kommunikation und soziales Verhalten. Der Referenzrahmen Schulqualität NRW bietet den Rahmen für die Entwicklung und Evaluation des Informatikunterrichts.

Überfachliche Grundsätze:

- 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 den aktuellen Entwicklungen der Informatik, wobei die Inhalte altersgemäß und didaktisch reduziert erarbeitet werden.
- 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.
- Der Unterricht ist problemorientiert und soll von realen Problemen ausgehen und sich auf solche rückbeziehen.
- Der Unterricht folgt dem Prinzip der Exemplarität 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 Schüler:innen an Bedeutsamkeit.
- Der Unterricht ist handlungsorientiert, d.h. projekt- und produktorientiert angelegt.
- Die fachlichen Inhalte werden anhand von authentischen, tragfähigen, gendersensiblen und motivierenden Problemstellungen gelehrt und gelernt.

4. Lehr und Lernmittel

Die Schule verwendet folgende Lehrwerke:

Unterstufe: *C.C.Buchner Informatik 5/6*

Einführungsphase: *Westermann Informatik 1*

Qualifikationsphase: *Westermann Informatik 2*

Darüberhinaus verwendet die Schule in der Klasse 5/6 die Programmierumgebung *Scratch* und nach Möglichkeit die *Lego-Spike-Roboter*.

In der Mittelstufe sollen *Calliopes* angeschafft werden.

In der Oberstufe wird als Programmierumgebung *Greenfoot* und *BlueJ* verwendet.

5. Grundsätze der Leistungsbewertung und Leistungsrückmeldung

Auf der Grundlage von §48 SchulG¹⁵, §6der APO-SI¹⁶ sowie Kapitel 3 des Kernlehrplans Informatik für die gymnasiale Oberstufe hat die Fachkonferenz der Alfred Krupp Schule 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.

Die Bewertung der Leistung berücksichtigt Lern- und Leistungssituationen. Einerseits soll dabei Schülerinnen und Schülern deutlich gemacht werden, in welchen Bereichen aufgrund des vorangegangenen Unterrichts stabile Kenntnisse erwartet und bewertet werden. Andererseits werden Fehler in neuen Lernsituationen im Sinne einer Fehlerkultur für den Lernprozess genutzt. Dabei sollen die Kompetenzbereiche zu gleichen Teilen in die Bewertung einfließen.

Die Leistungen im Unterricht werden in der Regel auf der Grundlage von Unterrichtshandlungen beurteilt. Darüberhinaus werden sowohl digitale als auch analoge Lernprodukte, wie z.B. Hefte, Portfolios, Dokumentationen, Präsentationen bewertet.

In der **Erprobungsstufe** wird Informatik als mündliches Fach unterrichtet, d.h. es werden keine Klausuren oder Klassenarbeiten geschrieben.

5.1 Beurteilungsbereich sonstige Mitarbeit:

¹⁵https://recht.nrw.de/lmi/owa/br_text_anzeigen?v_id=10000000000000000524

¹⁶<https://bass.schule.nrw/pdf/9607.pdf>

Die Bewertungskriterien für Leistungsbeurteilungen müssen den Schülerinnen und Schülern bekannt sein.
Das Erreichen der Kompetenzen ist zu überprüfen durch:

1. Beobachtungen der Schülerinnen und Schüler

Kriterien:

- arbeitet zielgerichtet und lässt sich nicht ablenken
- bringt seine individuellen Kompetenzen in den Arbeitsprozess ein
- nutzt Hard- und Software zielgerichtet
- erreicht das Ergebnis in der zur Verfügung stehenden Zeit
- kann sich in Diskussionen auf die Argumente der Mitschülerinnen und Mitschüler beziehen
- hält sich an vereinbarte Regeln
- kann eigene Meinungen begründet vertreten
- kann den eigenen Arbeitsprozess reflektieren und die Erkenntnisse umsetzen
- übt seine Funktion innerhalb der Gruppe verantwortungsvoll aus

2. Bewertung der Arbeitsprodukte

Kriterien:

- Ausführlichkeit
- Nachvollziehbarkeit
- Angemessene Verwendung der Fachsprache

3. Bewertung Schriftlicher Übungen

Lernerfolgsüberprüfung durch kurze schriftliche Übungen

Über die Anzahl der schriftlichen Übungen entscheidet die Lehrkraft, in der Regel wird am Ende einer Unterrichtsreihe entweder ein Unterrichtsprodukt oder eine schriftliche Übung bewertet.

Schriftliche Übung dauern ca. 20 Minuten und umfassen den Stoff der letzten ca. 2–5 Stunden.

4. 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.

5.2 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.

Stufe / Halbjahr	Anzahl Klausuren	Dauer Klausuren
EF.1	1 Klausur	90 Minuten
EF.2	1 Klausur	90 Minuten
Q1.1	2 Klausuren	100 Minuten
Q1.2	2 Klausuren	100 Minuten
Q2.1	2 Klausuren	160 Minuten
Q2.2	1 Klausur	Klausur unter Abiturbedingungen

Die Aufgabentypen, sowie die Anforderungsbereiche I-III sind entsprechend den Vorgaben in Kapitel 3 des Kernlehrplans SII¹⁷ zu beachten.

Kriterien

¹⁷ (Lehrplan SII, 2014)

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.

5.3 Grundsätze der Leistungsrückmeldung und Beratung

Es wird zu Beginn eines jeden Halbjahres sichergestellt, dass die Grundsätze der den Schülerinnen und Schülern bekannt sind. 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 Fach in der Qualifikationsphase.

6. 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. Um erkannten ungünstigen Entscheidungen schnellstmöglich entgegenwirken zu können, können nach Absprache innerhalb der Fachschaft Unterrichtsvorhaben verändert werden.

7. Antidiskriminierung

Im Fach Informatik soll der respektvolle, diskriminierungsfreie Umgang mit Technologien sowie der bewusste Einsatz von Informatikmethoden zur Förderung von Chancengleichheit und Vielfalt vermittelt werden. Die Schülerinnen und Schüler sollen sensibilisiert werden, welche Auswirkungen algorithmische Entscheidungen, Datenverarbeitung und Technologiegestaltung auf unterschiedliche Personengruppen haben können. Der Unterricht erfolgt grundsätzlich im Sinne der Gleichberechtigung aller Schüler:innen.

Inhaltliche Schwerpunkte

1. Bewusster Umgang mit Daten und Algorithmen

- Diskussion über algorithmische Voreingenommenheit (Bias) und ihre Ursachen. (Klasse 5/6 beim Thema KI)
- Fallbeispiele zu diskriminierenden Algorithmen und deren gesellschaftliche Auswirkungen. (Klasse 5/6 beim Thema KI)

2. Vielfalt und Repräsentation in der Informatik

- Sensibilisierung für unterschiedliche Bedürfnisse und Perspektiven bei der Entwicklung von Technologien.
- Förderung der gleichberechtigten Teilhabe aller Geschlechter, sozialer Gruppen und Minderheiten an der Informatik.
- Film *Hidden figures* im Differenzierungskurs Informatik

3. Digitale Ethik und Verantwortung

- Diskussion ethischer Fragestellungen im Zusammenhang mit KI, Automatisierung und Datenschutz. (alle Jahrgangsstufen)
- Kritische Analyse von Beispielen, in denen Technologien diskriminierende Wirkungen haben. (Unterrichtsvorhaben Q1-V)
- Förderung eines reflektierten Umgangs mit digitalen Werkzeugen im Sinne der Antidiskriminierung.

8. Literaturverzeichnis

APO GoSt. (2025). Von BASS: <https://bass.schule.nrw/9607.htm> abgerufen

APO SI. (kein Datum). *Schulministerium*. Von 2025:

https://www.schulministerium.nrw/sites/default/files/documents/APO_SI.pdf abgerufen

Frey, M. (25. 03 2025). *Greenfoot*. Von Wikipedia: <http://de.wikipedia.org/wiki/Greenfoot> abgerufen

Informatik, V. A. (2017). *APO-GOST*. Von Standardsicherung:

<https://www.standardsicherung.schulministerium.nrw.de/abitur-gost/fach.php?fach=15>,
ab Vorgaben Abitur 2017 abgerufen

Lehrplan SII. (2014). *Lehrpläne Informatik SII*. Von Standardsicherung:

http://www.standardsicherung.schulministerium.nrw.de/lehrplaene/upload/klp_SII/if/IF_a4.pdf abgerufen

Michael Kölling, I. U. (2025). *Greenfoot*. Von <https://www.greenfoot.org> abgerufen

Ministerium für Schule und Bildung . (2021). *Lehrplannavigator NRW*. Von Kernlehrplan Informatik 5/6 :

https://www.schulentwicklung.nrw.de/lehrplaene/lehrplan/260/si_kl5u6_if_klp_2021_07_01.pdf abgerufen

NGB. (2020). *Lerntheke Mars Rover*. Von Greenfoot:

https://ngb.schule/wiki/Lernpfad:Lerntheke_Marsrover/2 abgerufen

Oracle. (2025). *BlueJ*. Von <https://bluej.org/> abgerufen

Qualis. (2017). *Material zu zentralen Abiturprüfung Informatik*. Von Standardsicherung:

https://www.schulentwicklung.nrw.de/lehrplaene/upload/klp_SII/if/Dokumentation_ZA-IF_GK-LK_ab_2018_2021_12_22.pdf abgerufen

Qualis. (2020). *Implementation der Klassen für das Zentralabitur*. Von Standardsicherung:

https://www.schulentwicklung.nrw.de/lehrplaene/upload/klp_SII/if/MaterialZABI/2020-03-11_Implementationen_von_Klassen_fuer_das_Zentralabitur_ab_2018.zip abgerufen

Qualis. (kein Datum). *Lehrpläne SII*. Von Standardsicherung:

http://www.standardsicherung.schulministerium.nrw.de/lehrplaene/upload/klp_SII/if/IF_a4.pdf abgerufen

Schulministerium NRW. (2025). *Schulministerium*. Von Schulgesetz:

<https://www.schulministerium.nrw/schulgesetz-fuer-das-land-nordrhein-westfalen>
abgerufen