

Schulinternes Curriculum für das Fach Informatik

Stand: Februar 2024



Inhalt	Seite
1. Aufgaben und Ziele des Informatikunterrichts in der SI und SII	2
2. Entscheidungen zum Unterricht in der Sekundarstufe I	3
2.1. Hinweise	3
2.2. Übersichtsraster und Sequenzierungen der Unterrichtsvorhaben	5
2.3. Umsetzung des Medienkompetenzrahmens	12
3. Entscheidungen zum Unterricht in der Sekundarstufe II	14
3.1. Unterrichtsvorhaben	14
3.1.1. <i>Übersichtsraster Unterrichtsvorhaben</i>	15
3.1.2. <i>Konkretisierte Unterrichtsvorhaben</i>	23
3.2. Grundsätze der fachmethodischen und fachdidaktischen Arbeit	49
3.3. Grundsätze der Leistungsbewertung und Leistungsrückmeldung	51
3.3.1. <i>Allgemeines</i>	51
3.3.2. <i>Beurteilungsbereich „Klausuren“</i>	52
3.3.3. <i>Beurteilungsbereich „Sonstige Mitarbeit“</i>	53
4. Entscheidungen zu fach- und unterrichtsübergreifenden Fragen und Beitrag zur UNESCO-Schule	54
5. Qualitätssicherung und Evaluation	55
6. Anhang	55

1. Aufgaben und Ziele des Informatikunterrichts in der SI und SII

Das Fach Informatik wird in der Sekundarstufe I am Pius-Gymnasium als **Pflichtfach in der 6. Klasse** zweistündig unterrichtet. Die Schülerinnen und Schüler werden in diesem Schuljahr altersgemäß an Kerngebiete der Informatik wie Informatiksysteme, Information und Daten – Darstellungen und Codierungen, Algorithmen und das Programmieren herangeführt.

Ab der Jahrgangsstufe 9 wird Informatik als Teil der Fächerkombination MPI (Mathematik-Physik-Informatik) im **Wahlpflichtbereich II** (WP II) dreistündig angeboten. In der zweijährigen Laufzeit dieser WP II Kurse wird in altersstufengerechter Weise unter anderem auf Grundlagen der Algorithmik am Beispiel einer didaktischen Lernumgebung, auf die technische Informatik am Beispiel von Schaltwerken und Schaltnetzen eingegangen. Der Unterricht erfolgt dabei in enger Verzahnung mit Inhalten der Mathematik und Physik und wird zum Teil in Form von fächerverbindenden Projekten und in Kooperation mit außerschulischen Partnern gestaltet.

In der Sekundarstufe II bietet das Pius-Gymnasium für die eigenen Schülerinnen 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 MPI-Unterricht besucht haben, wird in Kursen der Einführungsphase besonderer Wert darauf gelegt, dass keine Vorkenntnisse aus dem MPI-Unterricht zum erfolgreichen Durchlaufen des Kurses erforderlich sind.

Der Unterricht der Sekundarstufe II wird mit Hilfe der Programmiersprache Java durchgeführt. In der Einführungsphase wird zur Einführung in die objektorientierte Programmierung die didaktische Lernumgebung *Greenfoot* gewählt, welche das Erstellen von grafischen Programmen erleichtert. Im 2. Halbjahr findet dann zunehmend die komplexere Entwicklungsumgebung *BlueJ* und der *JAVA-Editor* Einsatz im Unterricht

Durch projektartiges Vorgehen, offene Aufgaben und Möglichkeiten, Problemlösungen zu verfeinern oder zu optimieren, entspricht der Informatikunterricht der Oberstufe in besonderem Maße den Erziehungszielen, Leistungsbereitschaft zu fördern, ohne zu überfordern.

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

Zurzeit besteht die Fachschaft Informatik des Pius-Gymnasiums aus vier Lehrkräften für die Sekundarstufe I und zwei Lehrkräften für die Sekundarstufe II. Es existieren zwei Computerräume mit 30 bzw. 24 Computerarbeitsplätzen, vier Laptopwagen jeweils 16 Geräten und ein Medienzentrum, in dem den Schülerinnen und Schülern außerhalb des Unterrichts 5 Plätze zur Verfügung stehen. Alle Arbeitsplätze sind an das schulinterne Rechnernetz angeschlossen, so dass Schülerinnen und Schüler mit Hilfe eines Benutzernamens und ein Passwort Zugang zum zentralen Server der Schule haben. Von allen Arbeitsplätzen kann somit ein Zugriff auf ihre eigenen Daten, zur Recherche im Internet oder zur Bearbeitung schulischer Aufgaben, erfolgen.

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

2. Entscheidungen zum Unterricht in der Sekundarstufe I

2.1. Hinweise

Die Kompetenzerwartungen und inhaltlichen Schwerpunkte wurden entsprechend dem Kernlehrplan für die Sekundarstufe I Nordrhein-Westfalen vorgenommen. Die Inhaltsfelder und Kompetenzbereiche entsprechen den Ausführungen im Kernlehrplan.

Inhaltsfelder: Information und Daten; Algorithmen; Automaten und künstliche Intelligenz; Informatiksysteme; Informatik, Mensch und Gesellschaft

Ein Teil der Kompetenzen versteht sich als verpflichtend für die Schulform Gymnasium sowie als Vertiefungs- und Differenzierungsmöglichkeit für leistungsstarke Schülerinnen und Schüler an anderen Schulformen.

Übergeordnete Kompetenzbereiche:

<p>Argumentieren (A) Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • formulieren Fragen zu einfachen informatischen Sachverhalten, • äußern Vermutungen zu informatischen Sachverhalten auf der Basis von Alltagsvorstellungen oder Vorwissen, • erläutern mögliche Auswirkungen des Einsatzes von Informatiksystemen, • begründen die Auswahl eines Informatiksystems, • <i>bewerten ein Ergebnis einer informatischen Modellierung.</i> 	<p>Darstellen und Interpretieren (DI) Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • beschreiben einfache Darstellungen von informatischen Sachverhalten, • stellen informatische Sachverhalte in geeigneter Form dar, • interpretieren informatische Darstellungen.
<p>Modellieren und Implementieren (MI) Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • erstellen informatische Modelle zu gegebenen Sachverhalten, • implementieren informatische Modelle unter Verwendung algorithmischer Grundstrukturen, • überprüfen Modelle und Implementierungen. 	<p>Kommunizieren und Kooperieren (KK) Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • beschreiben einfache informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht, • <i>anstelle der vorherigen KE: erläutern informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht,</i> • kooperieren in verschiedenen Formen der Zusammenarbeit bei der Bearbeitung einfacher informatischer Probleme, • strukturieren gemeinsam eine Lösung für ein informatisches Problem, • dokumentieren gemeinsam ihren Ar-

	<p>beitsprozess und ihre Ergebnisse auch mithilfe digitaler Werkzeuge,</p> <ul style="list-style-type: none">• setzen bei der Bearbeitung einer informatischen Problemstellung geeignete digitale Werkzeuge zum kollaborativen Arbeiten ein.
--	--

2.2. Übersichtsraster und Sequenzierungen der Unterrichtsvorhaben

Nachfolgend wird ein Überblick über die Unterrichtsvorhaben in der Klasse 6 gegeben.

0 Grundlagen (ca. 2 Stunden)

Informatik 5/6	
0.1 Regeln für den Informatikraum 0.2 Computertastatur und Computermaus 0.3 Richtig Sitzen vor dem Bildschirm	Auch wenn der Kernlehrplan Informatik 5/6 es nicht explizit fordert, kann es Sinn machen, gewisse Grundlagen zum Arbeiten mit Computern mit den Lernenden zu thematisieren bzw. den Wissensstand der Lerngruppe diesbezüglich zu prüfen. Im Zeitalter der Smartphones und Tablets ist nicht zwangsläufig jeder Schülerin bzw. Schüler gewohnt, mit einem Computer zu arbeiten.

1 Informatiksysteme (ca. 8 Stunden)

Informatik 5/6	Kompetenzerwartungen und inhaltliche Schwerpunkte	Übergeordnete Kompetenzerwartungen
1.1 Das EVA-Prinzip 1.2 Projekt: Woraus besteht ein Computer? 1.3 Informatiksysteme, Hardware und Software 1.4 Benutzerkonten und sichere Passwörter 1.5 Das Speichern von Dokumenten und Dateien 1.6 Ordnung muss sein – Der Dateimanager 1.7 Ordnerstrukturen darstellen und an-	<p><u>Inhaltliche Schwerpunkte:</u></p> <ul style="list-style-type: none"> - Aufbau und Funktionsweise von Informatiksystemen - Anwendung von Informatiksystemen <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ➤ benennen Beispiele für (vernetzte) Informatiksysteme aus ihrer Erfahrungswelt (DI), ➤ benennen Grundkomponenten von (vernetzten) Informatiksystemen und beschreiben ihre Funktionen (DI), ➤ beschreiben das Prinzip der Eingabe, Verarbeitung und Ausgabe (EVA-Prinzip) als grundlegendes Prinzip der Datenverarbeitung 	<ul style="list-style-type: none"> - Argumentieren (A) - Modellieren und Implementieren (MI) - Darstellen und Interpretieren (DI) - Kommunizieren und Kooperieren (KK)

<p>passen</p> <p>1.8 Arten der Datenspeicherung</p>	<p>(DI),</p> <ul style="list-style-type: none"> ➤ vergleichen Möglichkeiten der Datenverwaltung hinsichtlich ihrer spezifischen Charakteristika (u. a. Speicherort, Kapazität, Aspekte der Datensicherheit) (A), ➤ setzen zielgerichtet Informatiksysteme zur Verarbeitung von Daten ein (MI), ➤ erläutern Prinzipien der strukturierten Dateiverwaltung (A), ➤ setzen Informatiksysteme zur Kommunikation und Kooperation ein (KK). 	
---	--	--

2 Information und Daten – Informationsgehalt von Daten und ihre Codierung (ca. 8 Stunden)

Informatik 5/6	Kompetenzerwartungen und inhaltliche Schwerpunkte	Übergeordnete Kompetenzerwartungen
<p>2.1 Kommunikation im Alltag und in der Informatik</p> <p>2.2 Arten der Codierung</p> <p>2.3 Bits und Bytes</p> <p>2.4 Binärzahlen</p> <p>2.5 Textcodierung – Der ASCII-Code</p>	<p><u>Inhaltliche Schwerpunkte:</u></p> <ul style="list-style-type: none"> - Daten und ihre Codierung - Informationsgehalt von Daten <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ➤ erläutern den Datenbegriff anhand von Beispielen aus ihrer Erfahrungswelt (A), ➤ erläutern den Zusammenhang und die Bedeutung von Information und Daten (A), ➤ stellen eine ausgewählte Information in geeigneter Form als Daten formalsprachlich oder graphisch dar (DI), ➤ nennen Beispiele für die Codierung von Daten aus ihrer Erfahrungswelt (DI), ➤ codieren und decodieren Daten unter Verwendung des Binärsystems (MI), 	<ul style="list-style-type: none"> - Argumentieren (A) - Modellieren und Implementieren (MI) - Darstellen und Interpretieren (DI) - Kommunizieren und Kooperieren (KK)

	<ul style="list-style-type: none"> ➤ interpretieren ausgewählte Daten als Information im gegebenen Kontext (DI), ➤ erläutern Einheiten von Datenmengen (A / KK), ➤ <i>vergleichen Datenmengen hinsichtlich ihrer Größe mithilfe anschaulicher Beispiele aus ihrer Lebenswelt (DI),</i> 	
--	---	--

3 Information und Daten – Verschlüsselungsverfahren (ca. 6 Stunden)

Informatik 5/6	Kompetenzerwartungen und inhaltliche Schwerpunkte	Übergeordnete Kompetenzerwartungen
<p>3.1 Geheimnisse bewahren mit Verschlüsselung</p> <p>3.2 Verschlüsselungsverfahren – Monoalphabetische Verschlüsselung</p> <p>3.3 Verschlüsselungsverfahren – Transposition</p> <p>3.4 Verschlüsselungsverfahren – Steganographie</p>	<p><u>Inhaltliche Schwerpunkte:</u></p> <ul style="list-style-type: none"> - Verschlüsselungsverfahren <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ➤ erläutern ein einfaches Transpositionsverfahren als Möglichkeit der Verschlüsselung (DI), ➤ <i>vergleichen verschiedene Verschlüsselungsverfahren unter Berücksichtigung von ausgewählten Sicherheitsaspekten (DI).</i> 	<ul style="list-style-type: none"> - Darstellen und Interpretieren (DI)

4 Algorithmen (ca. 8 Stunden)

Informatik 5/6	Kompetenzerwartungen und inhaltliche Schwerpunkte	Übergeordnete Kompetenzerwartungen
4.1 Beschreibung von Abläufen 4.2 Algorithmen im Alltag 4.3 Genaue Anweisungen und Abläufe 4.4 Beschreibungen abkürzen 4.5 Bedingte Anweisung und Verzweigung 4.6 Vom Algorithmus zum Programm	<p><u>Inhaltliche Schwerpunkte:</u></p> <ul style="list-style-type: none"> - Algorithmen und algorithmische Grundkonzepte <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ➤ formulieren zu Abläufen aus dem Alltag eindeutige Handlungsvorschriften (DI), ➤ <i>überführen Handlungsvorschriften in einen Programmablaufplan (PAP) oder ein Struktogramm (MI),</i> ➤ führen Handlungsvorschriften schrittweise aus (MI), ➤ identifizieren in Handlungsvorschriften Anweisungen und die algorithmischen Grundstrukturen Sequenz, Verzweigung und Schleife (MI), 	<ul style="list-style-type: none"> - Modellieren und Implementieren (MI) - Darstellen und Interpretieren (DI)

5 Programmieren mit einer visuellen Programmierumgebung (ca. 11 Stunden)

Informatik 5/6	Kompetenzerwartungen und inhaltliche Schwerpunkte	Übergeordnete Kompetenzerwartungen
5.1 Einführung in die Programmierung mit Scratch 5.2 Sequenzen von Anweisungen an ein Objekt 5.3 Reagieren auf Ereignisse 5.4 Wiederholung mit fester Anzahl 5.5 Schleifen mit Abbruchbedingungen 5.6 Verzweigungen 5.7 Variablen 5.8 Zielgerichtetes Testen von Programmen 5.9 Projekt: Ein Projekt planen und durchführen	<p><u>Inhaltliche Schwerpunkte:</u></p> <ul style="list-style-type: none"> - Implementation von Algorithmen <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ➤ implementieren Algorithmen in einer visuellen Programmiersprache (MI), ➤ <i>implementieren Algorithmen unter Berücksichtigung des Prinzips der Modularisierung (MI),</i> ➤ überprüfen die Wirkungsweise eines Algorithmus durch zielgerichtetes Testen (MI), ➤ <i>ermitteln durch die Analyse eines Algorithmus dessen Ergebnis (DI),</i> ➤ <i>bewerten einen als Quelltext, Programmablaufplan (PAP) oder Struktogramm dargestellten Algorithmus hinsichtlich seiner Funktionalität (A).</i> 	<ul style="list-style-type: none"> - Argumentieren (A) - Modellieren und Implementieren (MI) - Darstellen und Interpretieren (DI)

6 Automatisierung und künstliche Intelligenz (ca. 9 Stunden)

Informatik 5/6	Kompetenzerwartungen und inhaltliche Schwerpunkte	Übergeordnete Kompetenzerwartungen
6.1 Automaten im Alltag 6.2 Zustandsdiagramme 6.3 Projekt: Automaten mit Scratch 6.4 Künstliche Intelligenz in unserem Alltag 6.5 Entscheidungsbäume 6.6 Lernen durch Training 6.7 Neuronale Netze 6.8 Projekt: KI mit Scratch	<p><u>Inhaltliche Schwerpunkte:</u></p> <ul style="list-style-type: none"> - Aufbau und Wirkungsweise einfacher Automaten - Maschinelles Lernen mit Entscheidungsbäumen - <i>Maschinelles Lernen mit neuronalen Netzen</i> <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ➤ erläutern die Funktionsweise eines Automaten aus ihrer Lebenswelt (A), ➤ stellen Abläufe in Automaten graphisch dar (DI), ➤ benennen Anwendungsbeispiele künstlicher Intelligenz aus ihrer Lebenswelt (A), ➤ stellen das Grundprinzip eines Entscheidungsbaumes enaktiv als ein Prinzip des maschinellen Lernens dar (DI), ➤ <i>beschreiben die grundlegende Funktionsweise künstlicher neuronaler Netze in verschiedenen Anwendungsbeispielen (KK).</i> 	<ul style="list-style-type: none"> - Argumentieren (A) - Darstellen und Interpretieren (DI) - Kommunizieren und Kooperieren (KK)

7 Informatik, Mensch und Gesellschaft (ca. 8 Stunden)

Informatik 5/6	Kompetenzerwartungen und inhaltliche Schwerpunkte	Übergeordnete Kompetenzerwartungen
<p>7.1 Kleine und große Netzwerke – Das Internet</p> <p>7.2 Daten und Gefahren im Internet</p> <p>7.3 Schutz von Daten mit Hilfe von Informatiksystemen</p> <p>7.4 Wem gehören die Daten? – Rechte von Nutzern</p> <p>7.5 Verhalten und Umgang mit sozialen Netzwerken</p>	<p><u>Inhaltliche Schwerpunkte:</u></p> <ul style="list-style-type: none"> - Informatiksysteme in der Lebens- und Arbeitswelt - Datenbewusstsein - Datensicherheit und Sicherheitsregeln <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ➤ beschreiben an Beispielen die Bedeutung von Informatiksystemen in der Lebens- und Arbeitswelt (KK), ➤ benennen an ausgewählten Beispielen Auswirkungen des Einsatzes von Informatiksystemen auf ihre Lebens- und Erfahrungswelt (A/KK), ➤ <i>anstelle der vorherigen KE: erläutern an ausgewählten Beispielen Auswirkungen des Einsatzes von Informatiksystemen (A/KK),</i> ➤ beschreiben anhand von ausgewählten Beispielen die Verarbeitung und Nutzung personenbezogener Daten (DI), ➤ erläutern anhand von Beispielen aus ihrer Lebenswelt Nutzen und Risiken beim Umgang mit eigenen und fremden Daten auch im Hinblick auf Speicherorte (A), ➤ beschreiben Maßnahmen zum Schutz von Daten mithilfe von Informatiksystemen (A). 	<ul style="list-style-type: none"> - Argumentieren (A) - Kommunizieren und Kooperieren (KK)

2.3. Umsetzung des Medienkompetenzrahmens

In der folgenden Tabelle wird die Umsetzung des Medienkompetenzrahmens abgebildet:

Medienkompetenzen	Beispiele zur Umsetzung in Informatik 5/6
1. Bedienen und Anwenden – Digitale Werkzeuge	
1.1 Medienausstattung (Hardware)	<ul style="list-style-type: none"> ➤ 1.2 Projekt: Woraus besteht ein Computer? S.16f ➤ 1.3 Informatiksysteme, Hardware und Software S.18f/A1 und A2
1.3 Datenorganisation	<ul style="list-style-type: none"> ➤ 1.5 Das Speichern von Dokumenten und Dateien S.22/A2, Aufgabe 1 ➤ 1.6 Ordnung muss sein – Der Dateimanager S.25/A3, Aufgabe 1 und 2 ➤ 1.7 Ordnerstrukturen darstellen und anpassen S.26f/A1, A2, Aufgabe 1 und 2 ➤ 1.8 Arten der Datenspeicherung S.28f
1.4 Datenschutz und Informationssicherheit	<ul style="list-style-type: none"> ➤ 7.3 Schutz von Daten mit Hilfe von Informatiksystemen S.132f ➤ 7.4 Wem gehören die Daten? – Rechte von Nutzern S.134f
2. Informieren und Recherchieren	
2.1 Informationsauswertung	<ul style="list-style-type: none"> ➤ 2.1 Kommunikation im Alltag und in der Informatik S.37/Aufgabe 3
2.4 Informationskritik	<ul style="list-style-type: none"> ➤ 7.5 Verhalten und Umgang mit sozialen Medien S.136/A2
3. Kommunizieren und Kooperieren	
3.1 Kommunikations- und Kooperationsprozesse	<ul style="list-style-type: none"> ➤ 1.3 Informatiksysteme, Hardware und Software S.19/Aufgabe 1 ➤ 2.5 Textcodierung – Der ASCII-Code S.45/Aufgabe 2d ➤ 4.2 Algorithmen im Alltag S.69/Aufgabe 3b

3.4 Cybergewalt und - kriminalität	➤ 7.2 Daten und Gefahren im Internet	S.130f
4. Produzieren und Präsentieren		
4.1 Medienproduktion und Präsentation	➤ 2.1 Kommunikation im Alltag und in der Informatik	S.37/Aufgabe3
4.4 Rechtliche Grund- lagen	➤ 7.4 Wem gehören die Daten? – Rechte von Nutzern	S.134f
5. Analysieren und Reflektieren		
5.4 Selbstregulierte Mediennutzung	➤ 7.5 Verhalten und Umgang mit sozialen Medien	S.136/A1
6. Problemlösen und Modellieren		
6.2 Algorithmen er- kennen	➤ 4 Algorithmen	S.65 – 82
6.3 Modellieren und Programmieren	➤ 5 Programmieren mit einer visuellen Programmierungsumgebung	S.83 – 106
6.4 Bedeutung von Algorithmen	➤ 6.4 Künstliche Intelligenz in unserem Alltag	S.114f

3. Entscheidungen zum Unterricht in der Sekundarstufe II

3.1. 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. Die in den Tabellen der Konkretisierungsebene angegebenen Kapitel und Materialien beziehen sich auf das im Unterricht verwendete Lehrwerk: Informatik, Lehrwerk für die gymnasiale Oberstufe, Schöningh

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.

3.1.1. Übersichtsraster Unterrichtsvorhaben

Einführungsphase

Einführungsphase	
<p><u>Unterrichtsvorhaben E-I</u></p> <p>Thema: <i>Was macht Informatik? - Einführung in die Inhaltsfelder der Informatik</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Kommunizieren und Kooperieren - Darstellen und Interpretieren - Argumentieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Informatiksysteme - Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Einsatz, Nutzung und Aufbau von Informatiksystemen - Wirkung der Automatisierung <p>Zeitbedarf: 6 Stunden</p>	<p><u>Unterrichtsvorhaben E-II</u></p> <p>Thema: <i>Grundlagen der objektorientierten Analyse, Modellierung und Implementierung</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Modellieren - Implementieren - Darstellen und Interpretieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Daten und ihre Strukturierung - Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Objekte und Klassen - Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: 8 Stunden</p>

Einführungsphase	
<p><u>Unterrichtsvorhaben E-III</u></p> <p>Thema: <i>Algorithmische Grundstrukturen in Java</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Modellieren - Implementieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Objekte und Klassen - Syntax und Semantik einer Programmiersprache - Analyse, Entwurf und Implementierung einfacher Algorithmen <p>Zeitbedarf: 18 Stunden</p>	<p><u>Unterrichtsvorhaben E-IV</u></p> <p>Thema: <i>Das ist die digitale Welt! – Einführung in die Grundlagen, Anwendungsgebiete und Verarbeitung binärer Codierung</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Kommunizieren und Kooperieren - Darstellen und Interpretieren - Argumentieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Informatiksysteme - Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Binäre Codierung und Verarbeitung - Besondere Eigenschaften der digitalen Speicherung und Verarbeitung von Daten <p>Zeitbedarf: 8 Stunden</p>

Einführungsphase	
<p><u>Unterrichtsvorhaben E-V</u></p> <p>Thema: <i>Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand lebensnaher Anforderungsbeispiele</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Kommunizieren und Kooperieren - Darstellen und Interpretieren - Argumentieren - Modellieren - Implementieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Objekte und Klassen - Syntax, Semantik der Programmiersprache - Analyse, Entwurf und Implementierung einfacher Algorithmen <p>Zeitbedarf: 18 Stunden</p>	<p><u>Unterrichtsvorhaben E-VI</u></p> <p>Thema: <i>Such- und Sortieralgorithmen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Modellieren - Darstellen und Interpretieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Algorithmen - Daten und ihre Strukturierung <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Algorithmen zum Suchen und Sortieren - Analyse, Entwurf und Implementierung einfacher Algorithmen - Objekte und Klassen <p>Zeitbedarf: 9 Stunden</p>

Einführungsphase	
<p><u>Unterrichtsvorhaben E-VII</u></p> <p>Thema: <i>Leben in der digitalen Welt – Immer mehr Möglichkeiten und immer mehr Gefahren!?</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Kommunizieren und Kooperieren - Darstellen und Interpretieren - Argumentieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Informatiksysteme - Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Geschichte der automatischen Datenverarbeitung - Wirkungen der Automatisierung - Dateisystem <p>Zeitbedarf: 12 Stunden</p>	<p>Summe Einführungsphase: 77</p>

Qualifikationsphase (Q1 und Q2) – GRUNDKURS

Qualifikationsphase 1	
<p><u>Unterrichtsvorhaben Q1-I</u></p> <p>Thema: <i>Wiederholung und Vertiefung der objektorientierten Modellierung</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Modellieren - Darstellen und Interpretieren - Implementieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Daten und ihre Strukturierung - Algorithmen - Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Objekte und Klassen - Wirkungen der Automatisierung <p>Zeitbedarf: 14 Stunden</p>	<p><u>Unterrichtsvorhaben Q1-II</u></p> <p>Thema: <i>Organisation und Verarbeitung von Daten I – Modellierung und Implementierung von Anwendungen mit dynamischen und linearen Datenstrukturen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Modellieren - Implementieren - Darstellen und Interpretieren - Argumentieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Daten und ihre Strukturierung - Algorithmen - Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Objekte und Klassen - Syntax und Semantik einer Programmiersprache - Analyse, Entwurf und Implementierung von Algorithmen - Algorithmen in ausgewählten informatischen Kontexten - Wirkungen der Automatisierung <p>Zeitbedarf: 24 Stunden</p>

Qualifikationsphase 1	
<p><u>Unterrichtsvorhaben Q1-III</u></p> <p>Thema: <i>Algorithmen zum Suchen und Sortieren auf linearen Datenstrukturen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Darstellen und Interpretieren - Modellieren - Implementieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Algorithmen - Formale Sprachen und Automaten - Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Analyse, Entwurf und Implementierung von Algorithmen - Algorithmen in ausgewählten informatischen Kontexten - Syntax und Semantik einer Programmiersprache - Wirkungen der Automatisierung <p>Zeitbedarf: 22 Stunden</p>	<p><u>Unterrichtsvorhaben Q1-IV</u></p> <p>Thema: <i>Automaten und formale Sprachen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Darstellen und Interpretieren - Modellieren - Implementieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Formale Sprachen und Automaten - Informatiksysteme - Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Syntax und Semantik einer Programmiersprache - Endliche Automaten und Kellerautomaten - Grammatiken regulärer und kontextfreier Sprachen - Scanner, Parser und Interpreter für eine reguläre Sprache - Möglichkeiten und Grenzen von Automaten und formalen Sprachen - Einzelrechner und Rechnernetzwerke - Grenzen der Automatisierung <p>Zeitbedarf: 30 Stunden</p>
Summe Qualifikationsphase 1: 90 Stunden	

Qualifikationsphase 2

Unterrichtsvorhaben Q2-I

Thema:

Organisation und Verarbeitung von Daten II – Modellierung und Implementierung von Anwendungen mit dynamischen nicht-linearen Datenstrukturen

Zentrale Kompetenzen:

- Argumentieren
- Darstellen und Interpretieren
- Modellieren
- Implementieren
- 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: 26 Stunden

Unterrichtsvorhaben Q2-II

Thema:

Aufbau von und Kommunikation in Netzwerken

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Informatiksysteme
- Algorithmen

Inhaltliche Schwerpunkte:

- Einzelrechner und Rechnernetzwerke
- Sicherheit
- Nutzung von Informatiksystemen
- Analyse, Entwurf und Implementierung von Algorithmen
- Algorithmen in ausgewählten informatischen Kontexten

Zeitbedarf: 26 Stunden

Qualifikationsphase 2	
<p><u>Unterrichtsvorhaben Q2-III</u></p> <p>Thema: <i>Nutzung, Modellierung und Implementation von relationalen Datenbanken in Anwendungskontexten</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Modellieren - Implementieren - Darstellen und Interpretieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten - Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Datenbanken - Algorithmen in ausgewählten informatischen Kontexten - Syntax und Semantik einer Programmiersprache - Sicherheit <p>Zeitbedarf: 22 Stunden</p>	
Summe Qualifikationsphase 2: 74 Stunden	

3.1.2. Konkretisierte Unterrichtsvorhaben

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

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

<https://www.standardsicherung.schulministerium.nrw.de/cms/zentralabitur-gost/faecher/fach.php?fach=15%20> (aufgerufen: 22.07. 2022)

Einführungsphase

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

Die Schülerinnen und Schüler

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

Unterrichtsvorhaben EF-I

Thema: Was macht Informatik? – Einführung in die Inhaltsfelder der Informatik

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

Vorhabenbezogene Konkretisierung:

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

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

In einem letzten Schritt kann ausgehend von den Inhaltsfeldern das Schulnetzwerk in Ansätzen so analysiert werden, dass ein kompetenter Umgang mit diesem ermöglicht wird.

Zeitbedarf: 6 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Informatiksysteme und ihr genereller Aufbau</p> <ul style="list-style-type: none"> (a) Daten und ihre Strukturierung (b) Algorithmen (c) Formale Sprachen und Automaten (d) Informatiksysteme (e) Informatik, Mensch und Gesellschaft 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> - bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A) - nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D) 	<p>Kapitel 1 Was macht Informatik?</p> <p>Als Anschauungsmaterial bieten sich Navigationsgeräte und Landkarten an.</p>
<p>2. Der kompetente Umgang mit dem Schulnetzwerk</p> <ul style="list-style-type: none"> (a) Erstellen und Anlegen von Ordnerstrukturen (b) Sortieren von Dateien und Ordnern (c) Eingabe von Befehlen über Eingabeaufforderung (d) Einzelrechner und Netzwerk (e) Sicherheit und Datenschutz 		<p>Kapitel 1 Was macht Informatik?</p> <p>Interview mit dem Netzwerkadministrator, Benutzer- und Datenschutzbestimmungen der Schule</p>

Unterrichtsvorhaben EF-II

Thema: Grundlagen der objektorientierten Analyse, Modellierung und Implementierung

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

Vorhabenbezogene Konkretisierung:

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

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

Im Anschluss wird die objektorientierte Analyse für das Greenfoot-Szenario Planetenerkundung durchgeführt. Die vom Szenario vorgegebenen Klassen werden von Schülerinnen und Schülern in Teilen analysiert und entsprechende Objekte anhand einfacher Problemstellungen erprobt. Die Lernenden implementieren und testen einfache Programme. Die Greenfoot-Umgebung ermöglicht es, Beziehungen zwischen Klassen zu einem späteren Zeitpunkt (Kapitel 5) zu thematisieren. So kann der Fokus hier auf Grundlagen wie der Unterscheidung zwischen Klasse und Objekt, Attribute, Methoden, Objektidentität und Objektzustand gelegt werden.

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

Zeitbedarf: 8 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Identifikation von Objekten und Klassen</p> <p>(a) An einem lebensweltnahen Beispiel werden Objekte und Klassen im Sinne der objektorientierten Modellierung eingeführt.</p> <p>(b) Objekte werden durch Objektdiagramme, Klassen durch Klassendiagramme dargestellt.</p> <p>(c) Die Modellierungen werden einem konkreten Anwendungsfall entsprechend angepasst.</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> - ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften und ihre Operationen (M), - stellen den Zustand eines Objekts dar (D), - modellieren Klassen mit ihren Attributen und ihren Methoden (M), - implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), - implementieren Klassen in einer Programmiersprache, auch unter Nutzung dokumentierter Klassenbibliotheken (I). 	<p><i>Kapitel 2 Einführung in die Objektorientierung</i></p> <p>2.1 Objektorientierte Modellierung</p>
<p>2. Analyse von Objekten und Klassen im Greenfoot-Szenario</p> <p>(a) Schritte der objektorientierten Analyse, Modellierung und Implementation</p> <p>(b) Analyse und Erprobung der Objekte im Greenfoot-Szenario</p>		<p><i>Kapitel 2 Einführung in die Objektorientierung</i></p> <p>2.2 Das Greenfoot-Szenario „Planetenerkundung“</p> <p>Von der Realität zu Objekten</p> <p>Von den Objekten zu Klassen</p> <p>Klassendokumentation</p> <p>Objekte inspizieren</p> <p>Methoden aufrufen</p> <p>Objektidentität und Objektzustand</p>
<p>3. Implementierung einfacher Aktionen in Greenfoot</p> <p>(a) Quelltext einer Java-Klasse</p> <p>(b) Implementation eigener Methoden, Dokumentation mit JavaDoc</p> <p>(c) Programme übersetzen (Aufgabe des Compilers) und testen</p>		<p><i>Kapitel 2 Einführung in die Objektorientierung</i></p> <p>2.3 Programmierung in Greenfoot</p> <p>Methoden schreiben</p> <p>Programme übersetzen und testen</p>

Unterrichtsvorhaben EF-III

Thema: Algorithmische Grundstrukturen in Java

Leitfragen: Wie lassen sich Aktionen von Objekten flexibel realisieren?

Vorhabenbezogene Konkretisierung:

Das Ziel dieses Unterrichtsvorhabens besteht darin, das Verhalten von Objekten flexibel zu programmieren. Ein erster Schwerpunkt liegt dabei auf der Erarbeitung von Kontrollstrukturen. Die Strukturen Wiederholung und bedingte Anweisung werden an einfachen Beispielen eingeführt und anschließend anhand komplexerer Problemstellungen erprobt. Da die zu entwickelnden Algorithmen zunehmend umfangreicher werden, werden systematische Vorgehensweisen zur Entwicklung von Algorithmen thematisiert.

Ein zweiter Schwerpunkt des Unterrichtsvorhabens liegt auf dem Einsatz von Variablen. Beginnend mit lokalen Variablen, die in Methoden und Zählschleifen zum Einsatz kommen, über Variablen in Form von Parametern und Rückgabewerten von Methoden, bis hin zu Variablen, die die Attribute einer Klasse realisieren, lernen die Schülerinnen und Schüler die unterschiedlichen Einsatzmöglichkeiten des Variablenkonzepts anzuwenden.

Zeitbedarf: 18 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Algorithmen (a) Wiederholungen (While-Schleife, Zählschleife) (b) bedingte Anweisungen (c) Verknüpfung von Bedingungen durch die logischen Funktionen UND, ODER und NICHT (d) Systematisierung des Vorgehens zur Entwicklung von Algorithmen zur Lösung komplexerer Probleme	Die Schülerinnen und Schüler - analysieren und erläutern einfache Algorithmen und Programme (A), - entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M), - ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen zu (M), - modifizieren einfache Algorithmen und Programme (I), - implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I), - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), - implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), - testen Programme schrittweise anhand von Beispielen (I), - interpretieren Fehlermeldungen und korrigieren den Quellcode (I).	Kapitel 3 Algorithmen 3.1 Kontrollstrukturen 3.2 Wiederholungen 3.3 Zählschleifen 3.4 Bedingte Anweisungen 3.5 Logische Operationen 3.6 Algorithmen entwickeln
2. Variablen und Methoden (a) Implementierung eigener Methoden mit lokalen Variablen, auch zur Realisierung einer Zählschleife (b) Implementierung eigener Methoden mit Parameterübergabe und/oder Rückgabewert (c) Implementierung von Kontrollstrukturen (d) Realisierung von Attributen	(siehe oben)	Kapitel 4 Variablen und Methoden 4.1 Variablen 4.2 Methoden

Unterrichtsvorhaben EF-IV

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

Leitfragen: Wie werden binäre Informationen gespeichert und wie können sie davon ausgehend weiterverarbeitet werden? Wie unterscheiden sich analoge Medien und Geräte von digitalen Medien und Geräten? Wie ist der Grundaufbau einer digitalen Rechenmaschine?

Vorhabenbezogene Konkretisierung:

Das Unterrichtsvorhaben hat die binäre Speicherung und Verarbeitung sowie deren Besonderheiten zum Inhalt.

Im ersten Schritt erarbeiten die Schülerinnen und Schüler anhand ihnen bekannter technischer Gegenstände die Gemeinsamkeiten, Unterschiede und Besonderheiten der jeweiligen analogen und digitalen Version. Nach dieser ersten grundlegenden Einordnung des digitalen Prinzips wenden die Schülerinnen und Schüler das Binäre als Zahlensystem mit arithmetischen und logischen Operationen an und codieren Zeichen binär.

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

Zeitbedarf: 8 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Analoge und digitale Aufbereitung und Verarbeitung von Daten</p> <p>(a) Erarbeitung der Unterschiede von analog und digital</p> <p>(b) Zusammenfassung und Bewertung der technischen Möglichkeiten von analog und digital</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> - bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A) - stellen ganze Zahlen und Zeichen in Binärcodes dar (D) - interpretieren Binärcodes als Zahlen und Zeichen (D) 	<p><i>Die digitale Welt 001 – Von analog zu digital</i></p>
<p>2. Der Umgang mit binärer Codierung von Informationen</p> <p>(a) Das binäre (und hexadezimale) Zahlensystem</p> <p>(b) Binäre Informationsspeicherung</p> <p>(c) Binäre Verschlüsselung</p> <p>(d) Implementation eines Binärumrechners</p>	<ul style="list-style-type: none"> - beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A) - nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K) 	<p><i>Die digitale Welt 010 – Binäre Welt</i></p>
<p>3. Aufbau informatischer Systeme</p> <p>(a) Identifikation des EVA-Prinzips als grundlegende Arbeitsweise informatischer Systemen</p> <p>(b) Nachvollziehen der von-Neumann-Architektur als relevantes Modell der Umsetzung des EVA-Prinzips</p>	<ul style="list-style-type: none"> - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I) 	<p><i>Die digitale Welt 011 – Der Von-Neumann-Rechner</i></p>

Unterrichtsvorhaben EF-V

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 dieses dargestellt und realisiert?

Vorhabenbezogene Konkretisierung:

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

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

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

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

Zeitbedarf: 18 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Umsetzung von Anforderungen in Entwurfsdiagramme</p> <p>(a) Aus Anforderungsbeschreibungen werden Objekte mit ihren Eigenschaften identifiziert</p> <p>(b) Gleichartige Objekte werden in Klassen (Entwurf) zusammengefasst und um Datentypen und Methoden erweitert</p> <p>2. Implementationsdiagramme als erster Schritt der Programmierung</p> <p>(a) Erweiterung des Entwurfsdiagramms um Konstruktoren und get- und set-Methoden</p> <p>(b) Festlegung von Datentypen in Java, sowie von Rückgaben und Parametern</p> <p>(c) Entwicklung von Klassendokumentationen</p> <p>(d) Erstellung von Sequenzdiagrammen als Vorbereitung für die Programmierung</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> - analysieren und erläutern eine objektorientierte Modellierung (A), - stellen die Kommunikation zwischen Objekten grafisch dar (M), - ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), - modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), - 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), 	<p>Kapitel 5 Klassenentwurf</p> <p>5.1 Von der Realität zum Programm</p> <p>5.2 Objekte</p> <p>5.3 Klassen und Beziehungen entwerfen</p> <p>Kapitel 5 Klassenentwurf</p> <p>5.4 Klassen und Beziehungen implementieren</p>
<p>3. Programmierung anhand der Dokumentation und des Implementations- und Sequenzdiagrammes</p> <p>(a) Klassen werden in Java-Quellcode umgesetzt.</p> <p>(b) Das Geheimnisprinzip wird umgesetzt.</p> <p>(c) Einzelne Klassen und das Gesamtsystem werden anhand der Anforderungen und Dokumentationen auf ihre Korrektheit überprüft.</p> <p>4. Vererbungsbeziehungen</p> <p>(a) Das Grundprinzip der Vererbung wird erarbeitet.</p> <p>(b) Die Vorteile der Vererbungsbeziehungen</p> <p>(c) Vererbung wird implementiert</p>	<ul style="list-style-type: none"> - 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) 	<p>Kapitel 5 Klassenentwurf</p> <p>5.4 Klassen und Beziehungen implementieren</p> <p>Kapitel 5 Klassenentwurf</p> <p>5.5 Vererbung</p>
<p>5. Softwareprojekt</p> <p>(a) Analyse und Dekonstruktion eines Spiels (Modelle, Quelltexte)</p> <p>(b) Erweiterung des Spiels um weitere Funktionalitäten</p> <p>(c) Modellierung eines Spiels aufgrund einer Anforderungsbeschreibung, inklusive einer grafischen Benutzeroberfläche</p> <p>(d) (arbeitsteilige) Implementation des Spiels</p>		<p>Kapitel 7 Softwareprojekte</p> <p>7.1 Softwareentwicklung</p> <p>7.2 Oberflächen</p>

Unterrichtsvorhaben EF-VI

Thema: Such- und Sortieralgorithmen

Leitfragen: Wie können Objekte bzw. Daten effizient gesucht und sortiert werden?

Vorhabenbezogene Konkretisierung:

Dieses Unterrichtsvorhaben beschäftigt sich mit der Erarbeitung von Such- und Sortieralgorithmen. 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 lernen die Schülerinnen und Schüler das Feld als eine erste Datensammlung kennen. Alle Algorithmen dieses Kapitels arbeiten auf einem Feld.

Die Schülerinnen und Schüler lernen zunächst Strategien des Suchens (lineare Suche, binäre Suche, Hashing) und dann des Sortierens (Selection Sort, Insertion Sort, Bubble Sort) kennen.

Daran anschließend lernen die Schülerinnen und Schüler zunächst Strategien des Suchens (lineare Suche, binäre Suche, Hashing) und dann des Sortierens (Selection Sort, Insertion Sort, Bubble Sort) kennen. Die Projekteinstiege dienen dazu, die jeweiligen Strategien handlungsorientiert zu erkunden und intuitive Effizienzbetrachtungen der Suchalgorithmen vorzunehmen.

Schließlich wird die Effizienz unterschiedlicher Sortierverfahren beurteilt.

Zeitbedarf: 9 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Modellierung und Implementation von Datenansammlungen (a) Modellierung von Attributen als Felder (b) Deklaration, Instanziierung und Zugriffe auf ein Feld	Die Schülerinnen und Schüler - analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D) - entwerfen einen weiteren Algorithmus zum Sortieren (M) - beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeit und Speicherplatzbedarf (A) - ordnen Attributen lineare Datenansammlungen zu (M)	Kapitel 6 Suchen und Sortieren 6.1 Speichern mit Struktur – Arrays
2. Explorative Erarbeitung von Suchverfahren (a) Erkundung von Strategien für das Suchen auf unsortierten Daten, auf sortierten Daten und mithilfe einer Berechnungsfunktion (b) Vergleich der drei Verfahren durch intuitive Effizienzbetrachtungen		Kapitel 6 Suchen und Sortieren Projekteinstieg 1: Suchen 6.2 Suchen mit System Lineare Suche Binäre Suche Hashing
3. Systematisierung von Algorithmen und Effizienzbetrachtungen (a) Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode (b) Anwendung von Sortieralgorithmen auf verschiedene Beispiele (c) Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche (d) Analyse eines weiteren Sortieralgorithmus (sofern nicht in (a) bereits geschehen)		Kapitel 6 Suchen und Sortieren Projekteinstieg 2: Sortieren 6.3 Ordnung ist das halbe Leben!? – Sortieren Sortieren Selection Sort Insertion Sort Bubble Sort

Unterrichtsvorhaben EF-VII

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

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

Vorhabenbezogene Konkretisierung:

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

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

Ausgehend von dieser Betrachtung kann die aktuelle Informatik hinsichtlich ihrer Leistungsfähigkeit analysiert werden. Dabei soll herausgestellt werden, welche positiven und negativen Folgen Informatiksysteme mit sich bringen können.

Zeitbedarf: 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Schriftzeichen, Rechenmaschine, Computer</p> <p>(a) Anhand von Schwerpunkten, wie z.B. Datenspeicherung, Maschinen, Vernetzung sollen wichtige Entwicklungen der Informatik vorgestellt werden.</p> <p>(b) Anhand der unterschiedlichen Schwerpunkte sollen universelle Tendenzen der Entwicklung der Informationsverarbeitung erarbeitet werden.</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> - bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A), - erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A) - nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K) - erarbeiten selbständig Inhalte und stellen diese dem Plenum durch Referate in Form IT-gestützter Präsentationen vor (A) 	<p><i>Die digitale Welt 100 – Von der Schrift zum Smartphone</i></p>
<p>2. Die Informationsverarbeitung und ihre Möglichkeiten und Gefahren</p> <p>(a) Ausgehend von 1. werden Tendenzen der Entwicklung der Informatik erarbeitet</p> <p>(b) Informatik wird als Hilfswissenschaft klassifiziert, die weit über ihren originären Bereich hinaus Effizienz- und Leistungssteigerungen erzeugt</p> <p>(c) Anhand von Fallbeispielen werden technische und organisatorische Vorteile, sowie deren datenschutzrechtlichen Nachteile betrachtet.</p>		<p><i>Die digitale Welt 100 – Das Leben in der digitalen Welt</i></p>

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 und Vertiefung der objektorientierten Modellierung

Leitfragen: Wie wird aus einem anwendungsbezogenen Sachkontext ein informatisches Klassenmodell entwickelt? Wie werden Attribute, Methoden und Beziehungen identifiziert, den Klassen zugeordnet und dargestellt? Welche Auswirkungen hat die informatisch-technische Entwicklung auf das Leben der Menschen?

Vorhabenbezogene Konkretisierung:

Der bereits bekannte objektorientierte Zugang zu informatischer Modellierung wird von einer allgemeinen Betrachtung dieses informatischen Konzepts auf eine konkrete Problematik übertragen. Anhand dieser wird eine anwendungsbezogene Implementation Schritt für Schritt von der Objektidentifikation über das Entwurfs- und Implementationsdiagramm durchlaufen.

Grundlegende Modellierungskonzepte wie Sichtbarkeiten, Assoziationen, Vererbung sowie deren Darstellung in Entwurfs- und Klassendiagrammen und Dokumentationen werden wiederholt.

Ebenso wird erneut die grafische Darstellung von Objektkommunikation thematisiert.

Anhand von Gütekriterien und Eigenschaften von Modellierung entwickeln und bewerten die Schülerinnen und Schüler Klassenentwürfe.

Das Konzept der objektorientierten Modellierung wird um die Idee der abstrakten Klasse sowie um das Subtyping erweitert.

Zeitbedarf: 14 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Wiederholung der grundlegenden Konzepte der objektorientierten Programmierung</p> <p>a) Sichtweise der objektorientierten Informatik auf die Welt</p> <p>b) OOP als informatikspezifische Modellierung der Realität</p> <p>c) Schritte der Softwareentwicklung</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> - ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), - analysieren und erläutern objektorientierte Modellierungen (A), - modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), - ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), 	<p>Kapitel 1 Konzepte des objektorientierten Modellierens</p> <p>1.1 Modellierung der Realität 1.2 Die Welt ist voller Objekte Projekteinstieg: Klassentwurf – step by step</p>
<p>2. Erweiterung der objektorientierten Programmierung</p> <p>a) Umsetzung einer Anforderung in Entwurfs- und Klassendiagramm</p> <p>b) Objektkommunikation im Sequenzdiagramm</p> <p>c) Klassendokumentation</p> <p>d) Umsetzung von Teilen der Modellierung</p>	<ul style="list-style-type: none"> - modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), - verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie (M), - 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), 	<p>1.3 Gut geplant – Klassentwurf 1.4 Hierarchien machen´s einfacher – Vererbung</p>
<p>3. Mensch und Technik</p> <p>a) Informatiker verändern die Welt – Verantwortung der IT unter besonderer Berücksichtigung des katholischen Profils. Sollte alles Machbare auch getan werden?</p> <p>b) Automatisierung des Alltags durch Informatik</p> <p>– Der rasante Fortschritt der IT verändert alle Lebensbereiche: Informatik und Medizintechnik – längeres Leben Dank Maschine?</p>	<ul style="list-style-type: none"> - stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), - dokumentieren Klassen (D), - stellen die Kommunikation zwischen Objekten grafisch dar (D), - untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A), - untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A). 	<p>Die digitale Welt 001 – Mensch und Technik</p>

Unterrichtsvorhaben Q1-II

Thema:

Organisation und Verarbeitung von Daten I – Modellierung und Implementierung von Anwendungen mit dynamischen und linearen Datenstrukturen

Leitfragen:

Wie müssen Daten linear strukturiert werden, um in den gestellten Anwendungsszenarien eine beliebige Anzahl von Objekten verwalten zu können?

Vorhabensbezogene Konkretisierung:

Ausgehend von Alltagsbeispielen werden als Erstes die Anforderungen an eine Datenstruktur erschlossen. Anschließend werden die Möglichkeiten des Arrays untersucht, lineare Daten zu verwalten und über deren Grenzen/Probleme die Vorteile einer dynamischen linearen Struktur am Beispiel der Struktur Queue erarbeitet (Anwendungskontext Warteschlange). Die Klasse *Queue* selbst wird vorgegeben, die Operationen erläutert. Zur Vertiefung der Kenntnisse wird ein weiteres Anwendungsszenario eingeführt (Polizeikontrolle), dessen Lösung modelliert und implementiert wird. Darauf folgt die Erarbeitung der Struktur Stack, die mithilfe eines einfachen Anwendungsszenarios eingeführt (Biber/Palindrom) wird. Auch hier wird die Klasse *Stack* selbst vorgegeben und die Operationen erläutert. Weitere Aufgaben dienen der Vertiefung und Sicherung.

Um die Unterschiede der beiden Prinzipien FIFO und LIFO zu verstehen, werden zur Lösung der Aufgaben sowohl der Stack als auch die Queue benötigt.

Als letzte lineare dynamische Datenstruktur wird die Liste eingeführt. In dieser Sequenz liegt der Fokus auf der Möglichkeit, auf jedes Element zugreifen zu können. Nachdem die umfangreicheren Standardoperationen dieser Datenstruktur in einem einführenden Beispiel (Vokabeltrainer) erarbeitet und in einem weiteren Beispiel vertieft (LED) wurden, werden abschließend in einem Anwendungskontext verschiedene lineare Datenstrukturen angewendet. Die Modellierung erfolgt beim gesamten Vorhaben in Entwurfs- und Implementationsdiagrammen.

Zeitbedarf: 24 Stunden

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Die Datenstruktur Feld</p> <p>a) Erarbeitung der Anforderungen an eine Datenstruktur</p> <p>b) Wiederholung der Datenstruktur Array, Eigenschaften der Datenstruktur, Standardoperationen für ein und zweidimensionale Arrays</p> <p>c) Modellierung und Implementierung von Anwendungen</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> - erläutern Operationen dynamischer (linearer) Datenstrukturen (A), - implementieren Operationen dynamischer (linearer oder nichtlinearer) Datenstrukturen (I), - 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 Datensammlungen zu (M), 	<p>Kapitel 2 Lineare Datenstrukturen</p> <p>2.1 Anforderungen an eine Datenstruktur</p> <p>2.2 Datenansammlungen fester Größe – Arrays</p>
<p>2. Die Datenstruktur Schlange</p> <p>a) Modellierung und Implementierung der Verknüpfung von Objekten</p> <p>b) Generische Typen, Trennung von Verwaltung und Inhalt dyn. DS.</p> <p>c) Erläuterung von Problemstellungen, die nach dem FIFO-Prinzip bearbeitet werden</p> <p>d) Funktionalität der Schlange unter Verwendung der Klasse <i>Queue</i>; Erschließen der Standardoperationen</p> <p>e) Modellierung und Implementierung einer Anwendung auf der Basis einer Anforderungsbeschreibung mit Objekten der Klasse <i>Queue</i></p>	<ul style="list-style-type: none"> - interpretieren Fehlermeldungen und korrigieren den Quellcode (I), - stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D), - modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), - ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), - dokumentieren Klassen (D), - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), - verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie (M), 	<p>2.3 Wer zuerst kommt... – Schlangen</p>
<p>3. Die Datenstruktur Stapel</p> <p>a) Erläuterung von Problemstellungen, die nach dem LIFO-Prinzip bearbeitet werden</p> <p>b) Funktionalität der Klasse Stapel unter Verwendung der Klasse <i>Stack</i>, Erschließen der Standardoperationen</p> <p>c) Modellierung und Implementierung einer Anwendung auf Basis einer Anforderungsbeschreibung mit Objekten der Klasse <i>Stack</i></p> <p>d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (Objekte der Klassen <i>Queue</i>, <i>Stack</i> und <i>Array</i> (<i>Palindrom</i>))</p>	<ul style="list-style-type: none"> - untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A), - untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A). 	<p>2.4 Daten gut abgelegt – Stapel</p>
<p>4. Die Datenstruktur Liste</p> <p>a) Analyse der Möglichkeiten bisheriger Datenstrukturen zwecks</p>		<p>2.5 Flexibel für alle Fälle – lineare Listen</p>

<p>Bestimmung notwendiger Funktionalitäten für komplexere Anwendungen (Abgrenzung zu <i>Stack/Queue</i>, zusätzliche Fähigkeiten der Klasse <i>List</i>)</p> <p>b) Erarbeitung der Funktionalität der Liste unter Verwendung der Klasse <i>List</i></p> <p>c) Modellierung und Implementierung einer Anwendung mit Objekten der Klasse <i>List</i></p> <p>d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (<i>Stack</i>, <i>Queue</i>, <i>List</i>)</p>		
<p>5. Übungen und Vertiefungen zur Verwendung linearer und dynamischer Datenstrukturen anhand weiterer Problemstellungen</p>		<p>2.6 Prüfungsvorbereitung</p>
<p>6. Datenschutz</p> <p>a) Datenschutz als Grundrecht</p> <p>b) Das Datenschutzgesetz</p> <p>c) Datensammler</p>		<p><i>Projekteinstieg: Wartende Helden</i></p> <p><i>Mit dem Heldenspiel können alle im Kapitel behandelten Datenstrukturen erarbeitet werden. Das Spiel kann bis zu einem beliebigen Grad realisiert werden, sodass es sowohl als Einstieg als auch als ein umfassendes Projekt für lineare Datenstrukturen genutzt werden kann.</i></p> <p>Die digitale Welt 101 – Datenschutz</p>

Unterrichtsvorhaben Q1-III

Thema: Algorithmen zum Suchen und Sortieren auf linearen Datenstrukturen

Leitfragen: Nach welchen Grundprinzipien können Algorithmen strukturiert werden? Welche Qualitätseigenschaften sollten Algorithmen erfüllen? Wie können mithilfe von Such- und Sortieralgorithmen Daten in linearen Strukturen effizient (wieder-)gefunden werden?

Vorhabenbezogene Konkretisierung:

Zunächst werden anhand eines Anwendungsbeispiels übergreifende Algorithmeigenschaften (wie Korrektheit, Effizienz und Verständlichkeit) erarbeitet und Schritte der Algorithmusentwicklung wiederholt. Dabei kommen Struktogramme zur Darstellung von Algorithmen zum Einsatz.

Als besondere Struktur von Algorithmen wird die Rekursion an Beispielen veranschaulicht und gegenüber der Iteration abgegrenzt. Rekursive Algorithmen werden von den Schülerinnen und Schülern analysiert und selbst entwickelt.

In der zweiten Unterrichtssequenz geht es um die Frage, wie Daten in linearen Strukturen (lineare Liste und Array) (wieder-)gefunden werden können. Die lineare Suche als iteratives und die binäre Suche als rekursives Verfahren werden veranschaulicht und implementiert. Die Bewertung der Algorithmen erfolgt, indem jeweils die Anzahl der Vergleichsoperationen und der Speicherbedarf ermittelt wird.

Möchte man Daten effizient in einer linearen Struktur wiederfinden, so rückt zwangsläufig die Frage nach einer Sortierstrategie in den Fokus. Es wird mindestens ein iteratives und ein rekursives Sortierverfahren erarbeitet und implementiert sowie ihre Effizienz bewertet. Hier bietet sich die Implementation einer Testanwendung an, die die Laufzeiten unterschiedlicher Sortieralgorithmen misst. Daran anschließend folgen theoretische Analysen der Laufzeit.

Zeitbedarf: 22 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Eigenschaften von Algorithmen</p> <p>a) Qualitätseigenschaften von Algorithmen</p> <p>b) Strukturierung von Algorithmen mit Hilfe der Strategien „Modularisierung“ und „Teile und Herrsche“</p> <p>c) Analyse und Entwicklung von rekursiven Algorithmen</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> - analysieren und erläutern Algorithmen und Programme (A), - modifizieren Algorithmen und Programme (I), - stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D), - entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M), 	<p>Kapitel 3 Algorithmen</p> <p>3.1 Ohne Algorithmen läuft nichts</p> <p>3.2 Teile die Arbeit – rekursive Algorithmen</p>

<p>2. Suchen in Listen und Arrays</p> <p>a) Lineare Suche in Listen und Arrays</p> <p>b) Binäre Suche in einem Array</p> <p>c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf</p>	<ul style="list-style-type: none"> - implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), - testen Programme systematisch anhand von Beispielen und mithilfe von Testanwendungen (I), - implementieren und erläutern iterative und rekursive Such- und Sortierverfahren unterschiedlicher Komplexitätsklassen (Speicherbedarf und Laufzeitverhalten) (I), 	<p>3.3 Suchen – iterativ und rekursiv Projekteinstieg: Laufzeitanalyse experimentell</p>
<p>3. Sortieren auf Listen und Arrays</p> <p>a) Entwicklung und Implementierung eines iterativen Sortierverfahrens für eine Liste (Sortieren durch Einfügen)</p> <p>b) Entwicklung und Implementierung eines rekursiven Sortierverfahrens für eine Liste (Quicksort)</p> <p>c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf (praktische Tests und theoretische Analyse)</p> <p>d) Weitere Sortierverfahren auf Listen und Arrays (Sortieren durch Auswählen, Mergesort)</p> <p>4. Verantwortung der Informatik</p> <p>a) Der Unterschied zwischen Anwender und Produzent von Informatiksystemen</p> <p>b) Informatik und Ethik – Töten per Knopfdruck und Algorithmus: die Unvereinbarkeit taktischer Militärdrohnen mit christlichen Werten</p>	<ul style="list-style-type: none"> - beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A), - beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), - nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), - interpretieren Fehlermeldungen und korrigieren den Quellcode (I), - untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A), - untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A). 	<p>3.4 Sortieren – iterativ und rekursiv</p> <p>Implementation einer Testanwendung</p> <p>Die digitale Welt 011 – Verantwortung der Informatik</p>

Unterrichtsvorhaben Q1-IV

Thema: Automaten und formale Sprachen

Leitfragen:

Wie lassen sich reale Automaten durch ein Modell formal beschreiben? Wie kann die Art und Weise, wie ein Computer Zeichen (Eingaben) verarbeitet, durch Automaten dargestellt werden? Welche Eigenschaften besitzen Automaten und was können sie leisten? Wie werden sie dargestellt? Wie werden reguläre Sprachen durch eine Grammatik beschrieben? In welchem Verhältnis stehen endliche Automaten und Grammatiken? Welche Anwendungsfälle können durch endliche Automaten und Grammatiken regulärer Sprachen oder durch Kellerautomaten und kontextfreie Grammatiken beschrieben werden und welche nicht? 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? In welchen Anwendungen finden Automaten Einsatz?

Vorhabensbezogene Konkretisierung:

Ausgehend von der Beschreibung und Untersuchung realer Automaten wird das formale Modell eines endlichen Automaten entwickelt. Neben dem Mealy-Automaten geht es vor allem um den erkennenden endlichen Automaten. Auf die Erarbeitung der Beschreibung folgt die Modellierung eigener Automaten und die Untersuchung bestehender, um die Eigenschaften und Grenzen eines endlichen Automaten zu erkennen. Hierbei wird dessen Verhalten auf bestimmte Eingaben analysiert.

An den Themenkomplex *Endliche Automaten* schließt sich die Erarbeitung von Grammatiken regulärer Sprachen an. Die Untersuchung beginnt bei der Erschließung der formalen Beschreibung und wird mit der Entwicklung von Grammatiken zu regulären Sprachen fortgeführt. Hierbei wird auch die Beziehung von Grammatiken regulärer Sprachen zu endlichen Automaten an Beispielen erarbeitet und analysiert. Hierzu gehört auch die Untersuchung, welche Problemstellungen durch endliche Automaten und reguläre Grammatiken beschrieben werden können und welche nicht. Anhand einer Problemstellung wird die Grenze der endlichen Automaten aufgezeigt und in das Konzept des Kellerautomaten bzw. der kontextfreien Sprachen überführt.

Der Anwendungsbezug zu den theoretischen Grundlagen der Sprachen- und Automatentheorie wird durch das Modellieren und Implementieren von Scannern, Parsern und Interpretern vertieft.

Zeitbedarf: 30 Std.

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Endliche Automaten a) Erarbeitung der formalen Beschreibung eines Mealy-Automaten und der Darstellungsformen	Die Schülerinnen und Schüler - analysieren und erläutern die Eigenschaften endlicher Automaten und Kellerautomaten einschließlich ihres Verhaltens bei bestimmten Eingaben (A),	Kapitel 4 Endliche Automaten und formale Sprachen 4.1. Endliche Automaten Projekteinstieg: Schatzsuche

<p>b) Erarbeitung der formalen Beschreibung eines deterministischen endlichen Automaten (DEA) sowie dessen Darstellungsformen; Erschließung der Fachbegriffe Alphabet, Wort, (akzeptierte) Sprache, Determinismus</p> <p>c) Analyse der Eigenschaften von DEAs durch die Modellierung eines Automaten zu einer gegebenen Problemstellung, der Modifikation eines Automaten sowie die Überführung der gegebenen Darstellungsform in eine andere</p>	<ul style="list-style-type: none"> - ermitteln die Sprache, die ein endlicher Automat oder ein Kellerautomat akzeptiert (D), - entwickeln und modifizieren zu einer Problemstellung endliche Automaten oder Kellerautomaten (M), - stellen endliche Automaten in Tabellen und Graphen dar und überführen sie in die jeweils andere Darstellungsform (D), - entwickeln zur Grammatik einer regulären oder kontextfreien Sprache einen zugehörigen endlichen Automaten oder einen Kellerautomaten (M), - analysieren und erläutern Grammatiken regulärer und kontextfreier Sprachen (A), - modifizieren Grammatiken regulärer und kontextfreier Sprachen (M), 	
<p>2. Grammatiken regulärer Sprachen</p> <p>a) Erarbeitung der formalen Beschreibung einer regulären Grammatik (Sprache, Terminal und Nicht-Terminal, Produktionen und Produktionsvorschriften)</p> <p>b) Analyse der Eigenschaften einer regulären Grammatik durch deren Entwicklung und Modellierung zu einer gegebenen Problemstellung.</p>	<ul style="list-style-type: none"> - ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A), - entwickeln zu einer regulären oder kontextfreien Sprache eine Grammatik, die die Sprache erzeugt (M), - entwickeln zur akzeptierten Sprache eines Automaten eine zugehörige Grammatik (M), - beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D), - erläutern die Grenzen endlicher Automaten und regulärer Sprachen im Anwendungszusammenhang (A), 	<p>4.2 Formale Sprachen</p>
<p>3. Scanner, Parser, Interpreter</p> <p>a) Erarbeitung der Konzepte von Scanner, Parser und Interpreter</p> <p>b) Modellierung und Implementation von Scanner, Parser und Interpreter</p>	<ul style="list-style-type: none"> - modellieren und implementieren Scanner, Parser und Interpreter zu einer gegebenen regulären Sprache (I), - erläutern die Ausführung eines einfachen maschinennahen Programms sowie die Datenspeicherung auf einer „Von-Neumann-Architektur“ (A), 	<p>4.3 Vom Text zum Programm</p> <p>4.4 Scanner und Parser</p>
<p>4. Übungen und Vertiefungen</p> <p>Verwendung endlicher Automaten und Grammatiken regulärer Sprachen</p> <p>5. Grundsätzliche Arbeitsweise eines Computers und Grenzen der Berechenbarkeit</p> <p>a) Von-Neumann-Architektur und die Ausführung maschinennaher Programme</p> <p>b) Grenzen der Berechenbarkeit anhand des Halteproblems, nicht effizient berechenbare Probleme</p>	<ul style="list-style-type: none"> - untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A). 	<p>4.5 Prüfungsvorbereitung</p> <p>Die digitale Welt 101 – Maschinennahe Programmierung</p> <p>Die digitale Welt 100 – Berechenbarkeit</p>

Unterrichtsvorhaben Q2-I

Thema: Organisation und Verarbeitung von Daten II – Modellierung und Implementierung von Anwendungen mit dynamischen nicht-linearen Datenstrukturen

Leitfragen: Wie können Daten mithilfe von Baumstrukturen verwaltet werden? Wie können mit binären Suchbäumen Inhalte sortiert verwaltet werden und welche Vor- und Nachteile bietet dies?

Vorhabenbezogene Konkretisierung:

Anhand des Anwendungskontextes Spielbäume werden zunächst der generelle Aufbau von Baumstrukturen (auch nicht-binäre) und wichtige Grundbegriffe erarbeitet. Die Darstellung von Bäumen mit Knoten und Kanten wird eingeführt.

Anschließend rückt der Fokus auf die binären Bäume, deren rekursiver Aufbau für die Traversierung der Datenstruktur genutzt wird. Die Preorder-Traversierung wird verwendet, um einen gespeicherten Inhalt in einem Binärbaum zu finden (Tiefensuche).

Der Anwendungskontext Ahnenbaum wird mithilfe der Klasse *BinaryTree* (der Materialien für das Zentralabitur in NRW) modelliert und (ggf. in Teilen) implementiert. Dabei wird u. a. die Erzeugung eines Binärbaums mithilfe der beiden Konstruktoren der Klasse *BinaryTree* thematisiert.

Möchte man Daten geordnet speichern, so bietet sich die Struktur des binären Suchbaums an. An Beispielen wird zunächst das Prinzip des binären Suchbaums erarbeitet. Die Operationen des Suchens, Einfügens, Löschens und der sortierten Ausgabe werden thematisiert.

Um Daten in einem Anwendungskontext mithilfe eines binären Suchbaums verwalten zu können, müssen sie in eine Ordnung gebracht werden können, d. h. sie müssen vergleichbar sein. Diese Vorgabe wird mithilfe des Interfaces *ComparableContent* realisiert, das alle Klassen, dessen Objekte in einem Suchbaum verwaltet werden sollen, implementieren müssen. Auf diese Weise wird ein Anwendungskontext (Benutzerverwaltung) mithilfe der Klassen *BinarySearchTree* und *ComparableContent* modelliert und (ggf. in Teilen) implementiert.

Zeitbedarf: 26 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Aufbau von Baumstrukturen und Grundbegriffe</p> <p>a) Erarbeitung der Begriffe Wurzel, Knoten, Blatt, Kante, Grad eines Knotens und eines Baumes, Pfad, Tiefe, Ebene, Teilbaum</p> <p>b) Aufbau und Darstellung von Baumstrukturen in verschiedenen Anwendungskontexten</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> - stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D), - erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), - implementieren Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (I), - analysieren und erläutern Algorithmen und Programme (A), - stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D). 	<p>Kapitel 5 Nicht-lineare Datenstrukturen</p> <p>5.1 Spielen mit Struktur – Baumstrukturen</p> <p>Projekteinstieg 1: Spielbäume</p>

<p>2. Binäre Bäume</p> <p>a) rekursiver Aufbau eines binären Baums</p> <p>b) Traversierungen (pre-, in-, postorder)</p> <p>c) Modellierung eines Binärbaums in einem Anwendungskontext mit Hilfe der Klasse BinaryTree (als Entwurfs- und Implementationsdiagramm)</p> <p>d) Implementation einer Anwendung der Datenstruktur binärer Baum (ggf. in Teilen)</p>	<ul style="list-style-type: none"> - beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), - ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), - ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), - modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), - verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M), 	<p>5.2 Zwei Nachfolger sind genug! - Binäre Bäume</p> <p>Implementation des Projekts Ahnenbaum</p>
<p>3. Binäre Suchbäume</p> <p>a) Prinzip des binären Suchbaums, Ordnungsrelation</p> <p>b) Operationen auf dem binären Suchbaum (Suchen, Einfügen, Löschen, sortierte Ausgabe)</p> <p>c) Modellierung eines binären Suchbaums in einem Anwendungskontext mit Hilfe der Klasse BinarySearchTree (als Entwurfs- und Implementationsdiagramm) und dem Interface Item</p> <p>d) Implementation einer Anwendung der Datenstruktur binärer Suchbaum (ggf. in Teilen)</p>	<ul style="list-style-type: none"> - entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“, „Teilen und Herrschen“ und „Backtracking“ (M), - implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), - modifizieren Algorithmen und Programme (I), - nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), - interpretieren Fehlermeldungen und korrigieren den Quellcode (I), - testen Programme systematisch anhand von Beispielen (I). 	<p>5.3 Wer Ordnung hält, spart Zeit beim Suchen – Binäre Suchbäume</p> <p>Projekteinstieg 2: Binäre Suchbäume</p> <p>Implementation des Projekts Benutzerverwaltung</p>
<p>4. Graphen (fakultativ im GK)</p> <p>a) Anwendungskontexte für Graphen, Grundbegriffe</p> <p>b) Modellierung von Graphen mit Adjazenzlisten mit Hilfe der Klassen Graph und GraphNode und Adjazenzmatrizen.</p> <p>c) Algorithmus von Dijkstra</p> <p>d) Tiefen- und Breitensuche</p>		<p>Projekteinstieg 3: Navigationssysteme</p> <p>5.4 Navigieren mit Struktur – Graphen</p> <p>5.5 Modellierung von Graphen</p> <p>5.6 Kürzester Weg – der Algorithmus von Dijkstra</p> <p>5.7 Graphen durchsuchen</p>
<p>5. Übung und Vertiefungen der Verwendung von Binärbäumen, binären Suchbäumen (und Graphen) anhand weiterer Problemstellungen</p>		<p>5.8 Prüfungsvorbereitung</p>

Unterrichtsvorhaben Q2-II

Thema: Aufbau von und Kommunikation in Netzwerken

Leitfragen: Was macht menschliche Kommunikation aus? Welchen Stellenwert haben technische/informatische Hilfsmittel für die Kommunikation? Wie werden Daten in einem Netzwerk zwischen den Kommunikationspartnern übertragen? Wie ist die Arbeitsteilung in Netzwerken gestaltet? Wie kann sicher in Netzwerken kommuniziert werden? Wie werden Protokolle zur zuverlässigen Kommunikation eingesetzt und in Software umgesetzt?

Vorhabenbezogene Konkretisierung:

Ausgehend von alltäglicher Face-to-Face-Kommunikation werden die Grundprinzipien sowie die Bewertungskriterien von Kommunikation erläutert. Das Netzwerk wird als vorteilhafte Kommunikationsstruktur dargestellt und anhand von Topologien und Reichweiten kategorisiert. Ausgehend davon wird der Protokollbegriff entwickelt und anhand des TCP/IP-Schichtenmodells analysiert. Anschließend wird das Client-Server-Prinzip vorgestellt und in einer Projektarbeit angewandt. Es werden eigenständig Protokolle entwickelt und in einer Client-Server-Anwendung umgesetzt. Sichere Kommunikation in Netzen ist nur dank kryptografischer Verfahren möglich. Stellvertretend werden zwei symmetrische und ein asymmetrisches Verfahren erläutert, angewandt und bewertet.

Zeitbedarf: 26 Stunden

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Technische Kommunikation als Fortführung natürlicher Kommunikation</p> <p>a) Kommunikation im Shannon-Weaver-Modell</p> <p>b) Kriterien von technischen Kommunikationsarten</p> <p>c) Die Geschichte der technischen Kommunikation</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> - beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A), - analysieren und erläutern Protokolle zur Kommunikation in einem Client-Server-Netzwerk (A), - entwickeln und erweitern Protokolle zur Kommunikation in einem Client-Server-Netzwerk (M), 	<p>Kapitel 6 Kommunikation in Netzwerken</p> <p>6.1 Menschen kommunizieren – ohne und mit Technik</p> <p>Projekteinstieg: Kommunikation im Wilden Westen</p>
<p>2. Aufbau von Netzwerken und Kommunikationsregeln</p> <p>a) Das Netzwerk als Organisationsprinzip der Kommunikation und Möglichkeiten der Ausformung</p> <p>b) Geregelte technische Kommunikation durch Protokolle in Schichtenmodellen</p>	<ul style="list-style-type: none"> - erläutern das Prinzip der Nebenläufigkeit (A), - analysieren und erläutern Algorithmen und Methoden zur Client-Server-Kommunikation (A), - entwickeln und implementieren Algorithmen und Methoden zur Client-Server-Kommunikation (I), - analysieren und erläutern Eigenschaften, Funktionsweisen und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A), 	<p>6.2 Ohne Protokoll läuft nichts – Netzwerke</p>
<p>3. Aufgabenteilung in Netzwerken durch Server und Client</p> <p>a) Aufbau und Aufgaben der Client-Server-Struktur</p> <p>b) Protokolle zwischen Client und Server</p> <p>c) Implementation von Anwendungen unter Berücksichtigung der Client-Server-Struktur und gegebener Protokolle</p>	<ul style="list-style-type: none"> - nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zur Erschließung, Aufbereitung und Präsentation fachlicher Inhalte (D). 	<p>6.3 Einer für alle – Client-Server-Struktur</p> <p>Projektarbeit (z.B. Chatanwendung, Spiele wie Schiffe versenken oder Stein, Schere, Papier, Fussballmanager)</p>
<p>4. Kryptologie</p> <p>a) Veranschaulichen und Anwenden von symmetrischen und asymmetrischen kryptographischen Verfahren (Caesar, Vigenère, RSA, Diffie-Hellman Schlüsselaustausch)</p> <p>b) Bewertung der Verfahren hinsichtlich ihrer Sicherheit und ihrem Aufwand</p>		<p>Die digitale Welt 111 – Kryptologie</p>
<p>5. Übung und Vertiefung des Aufbaus von und der Kommunikation in Netzwerken</p>		<p>6.4 Prüfungsvorbereitung</p>

Unterrichtsvorhaben Q2-III

Thema: Nutzung und Modellierung und Implementation von relationalen Datenbanken in Anwendungskontexten

Leitfragen: Was sind Datenbanken und wie kann man mit ihnen arbeiten? Wie entwickelt man selbst eine Datenbank für einen Anwendungskontext?

Vorhabenbezogene Konkretisierung:

Am Beispiel eines Online-Buchhandels wird der Aufbau einer Datenbank sowie wichtige Grundbegriffe erarbeitet. Die Schülerinnen und Schüler nehmen dabei zunächst die Sicht der Anwender ein, die eine bestehende Datenbank beschreiben und analysieren und mithilfe von SQL-Abfragen Daten gezielt herausfiltern.

Mithilfe des Projekteinstiegs „Tabellen“ können bereits zu einem frühen Zeitpunkt des Unterrichtsvorhabens Redundanzen, Inkonsistenzen und Anomalien problematisiert werden.

Nachdem die Lernenden in der ersten Sequenz mit Datenbanken vertraut gemacht wurden, nehmen sie nun die Rolle der Entwickler an, indem sie selbst Datenbanken von Grund auf modellieren und das Modell in ein Relationenschema überführen. Sie arbeiten mit Entity-Relationship-Diagrammen, um Entitäten, zugehörige Attribute, Relationen und Kardinalitäten in Anwendungskontexten darzustellen. Gegebene ER-Diagramme werden analysiert, erläutert und modifiziert.

Der bereits in der ersten Sequenz problematisierte Begriff der Redundanz wird am Ende des Unterrichtsvorhabens wieder aufgegriffen, um die Normalisierung von Datenbanken zu thematisieren. Bestehende Datenbankschemata werden hinsichtlich der 1. bis 3. Normalform untersucht und (soweit nötig) normalisiert.

Den Abschluss des Unterrichtsvorhabens bildet eine Projektarbeit, bei der eine Datenbankanwendung von den Schülerinnen und Schülern modelliert, implementiert, dokumentiert und getestet wird. Die Implementation der Java-Anwendung wird dabei mit den Klassen *DatabaseConnector* und *QueryResult* (der Materialien für das Zentralabitur NRW) realisiert. Die Projektarbeit vertieft insbesondere die Kompetenzen im Bereich Kommunizieren und Kooperieren, da die Lernenden eigenverantwortlich und kooperativ arbeiten, ihre Arbeitsabläufe bewerten und Ergebnisse vorstellen.

Zeitbedarf: 22 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Nutzung von relationalen Datenbanken a) Aufbau von Datenbanksystemen und Grundbegriffe - Aufgaben und Eigenschaften eines Datenbanksystems - Erarbeitung der Begriffe Tabelle, Attribut, Attributwert, Datensatz, Datentyp, Primärschlüssel,	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), - verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I),	Kapitel 8 Datenbanken 7.1 Wissen speichern und verwalten – Datenbanksysteme 7.2 Daten anordnen mit Tabellen Projekteinstieg: Tabellen 7.3 Daten filtern mit SQL

<p>Datenbankschema</p> <ul style="list-style-type: none"> - Problematisierung von Redundanzen, Anomalien und Inkonsistenzen <p>b) SQL-Abfragen</p> <ul style="list-style-type: none"> - Erarbeitung der grundlegenden Sprachelemente von SQL (SELECT(DISTINCT), FROM, WHERE, JOIN) - Analyse und Erarbeitung von SQL-Abfragen (AND, OR, NOT, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, <>, >, <, >=, <=, LIKE, BETWEEN, IN, IS NULL, geschachtelte Select-Ausdrücke) <p>c) Vertiefung an einem weiteren Datenbankbeispiel</p>	<ul style="list-style-type: none"> - ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D), - ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M), - stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D), - modifizieren eine Datenbankmodellierung (M), - modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M), - bestimmen Primär- und Sekundärschlüssel (M), - implementieren eine relationales Datenbankschema als Datenbank (I), - analysieren und erläutern eine Datenbankmodellierung (A), - erläutern die Eigenschaften normalisierter Datenbankschemata (A), - überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D), 	<p>7.4 Komplexe Filter</p>
<p>2. Modellierung von relationalen Datenbanken</p> <p>a) Datenbankentwurf durch ER-Diagramme</p> <ul style="list-style-type: none"> - Ermittlung von Entitäten, zugehörigen Attributen, Beziehungen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms - Erläuterung und Erweiterung einer Datenbankmodellierung <p>b) Entwicklung eines relationalen Modells aus einem Datenbankentwurf</p> <ul style="list-style-type: none"> - Überführung eines Entity-Relationship-Diagramms in ein relationales Datenbankschema inklusive der Bestimmung von Primär- und Fremdschlüsseln <p>c) Normalformen</p> <ul style="list-style-type: none"> - Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten) 	<ul style="list-style-type: none"> - überführen Datenbankschemata in die 1. bis 3. Normalform (M), - nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung von Daten, zur Organisation von Arbeitsabläufen sowie zur Verteilung und Zusammenführung von Arbeitsanteilen (K), - wenden didaktisch orientierte Entwicklungsumgebungen zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I), - entwickeln mit didaktisch orientierten Entwicklungsumgebungen einfache Benutzungsoberflächen zur Kommunikation mit einem Informatiksystem (M). 	<p>7.5 Datenbankentwurf</p> <p>7.6 Umsetzung des ER-Modells</p> <p>Wiederaufgriff des Projekteinstiegs</p> <p>7.7 Datenbanken verbessern durch Normalformen</p>

<p>3. Implementation von relationalen Datenbanken Modellierung, Implementation, Dokumentation und Test einer Datenbankanwendung in einem Anwendungskontext (Projektarbeit)</p>		<p>7.3 Daten filtern mit SQL 7.4 Komplexe Filter 7.6 Umsetzung des ER-Modells Projektarbeit (z.B. Kassensystem, Cocktailverzeichnis, Umfragetool für die Abizeitung), ggf. auch mit Netzwerkfunktionalität</p>
<p>4. Übung und Vertiefung der Nutzung und Modellierung von relationalen Datenbanken</p>		<p>7.8 Prüfungsvorbereitung</p>

Unterrichtsvorhaben Q2-IV

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

3.2. Grundsätze der fachmethodischen und fachdidaktischen Arbeit

In Absprache mit der Lehrerkonferenz sowie unter Berücksichtigung des Schulprogramms hat die Fachkonferenz Informatik des Pius-Gymnasiums die folgenden fachmethodischen und fachdidaktischen Grundsätze beschlossen. In diesem Zusammenhang beziehen sich die Grundsätze 1 bis 14 auf fächerübergreifende Aspekte, die auch Gegenstand der Qualitätsanalyse sind, die Grundsätze 16 bis 22 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.
- 15) Der Unterricht beachtet die christlichen Wertvorstellungen einer bischöflichen Schule, wie sie im Schulprogramm festgelegt sind.

Fachliche Grundsätze:

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

3.3. Grundsätze der Leistungsbewertung und Leistungsrückmeldung

3.3.1. Allgemeines

Die Grundsätze der Leistungsbewertung ergeben sich aus dem Schulgesetz, das Verfahren aus der Verordnung über den Bildungsgang und die Abiturprüfung in der gymnasialen Oberstufe (APO-GOST).

Bewertet werden der Umfang der Kenntnisse, die methodische Selbstständigkeit in ihrer Anwendung sowie die sachgemäße schriftliche und mündliche Darstellung.

Bei der schriftlichen und mündlichen Darstellung wird auf sachliche und sprachliche Richtigkeit, auf fachsprachliche Korrektheit, auf gedankliche Klarheit und auf eine der Aufgabenstellung angemessene Ausdrucksweise geachtet.

Bei Gruppenarbeiten muss die individuelle Leistung des einzelnen Schülers erkennbar und bewertbar sein.

Dem Beurteilungsbereich „Sonstige Mitarbeit“ kommt der gleiche Stellenwert zu wie dem Beurteilungsbereich „Klausuren“. Die Teilnoten in den beiden Beurteilungsbereichen werden unabhängig voneinander gebildet. Die Gesamtnote wird nicht rein arithmetisch, sondern nach pädagogischen Gesichtspunkten gebildet. Dabei ist die Entwicklung der Schülerin bzw. des Schülers über einen längeren Zeitraum zu beachten.

Der Pädagogischer Leistungsbegriff

Die im Folgenden genannten Bedingungen sind Konsequenzen eines pädagogischen Leistungsbegriffs innerhalb der Fachschaft Informatik, der die Grundlage der Planung des Fachunterrichts am Pius-Gymnasium und der entsprechenden Leistungsbewertung bildet:

- Leistung setzt unter allen Beteiligten (also unter Schülerinnen und Schülern, Eltern, Lehrerinnen und Lehrern und anderweitig in der Schule involvierten) eine vertrauensvolle Beziehung voraus – sie fördert ebenso echte Lernprozesse, wie sie eine faire Leistungsmessung ermöglicht.
- Leistung setzt institutionalisierte und systematische Unterstützungsangebote voraus – nur so besteht eine echte Chance, individuelle Problemfelder wirklich bearbeiten zu können.
- Leistung setzt differenzierte Anregung voraus - denn Lernen und Leisten sind immer individuelle Prozesse.
- Leistung setzt eine vielfältige Basis der Leistungsbewertung voraus – nicht nur das Produkt, sondern auch Prozess und Präsentation müssen ebenso eine Rolle spielen, wie kreative, soziale und kognitive Handlungen.
- Leistung setzt eine regelmäßige Verständigung auf Augenhöhe voraus – wenn Leistung ein soziales Phänomen ist, muss man sie gemeinsam definieren, damit alle Beteiligten die Kriterien der Leistung und ihrer Messung annehmen können.

Erfolgreiches Lernen ist kumulativ und als ansteigende Progression zu verstehen. Dies erfordert, dass Lernerfolgsüberprüfungen darauf ausgerichtet sein müssen, Schülerinnen und Schülern Gele-

genheit zu geben, Kompetenzen, die sie in den vorangegangenen Jahren erworben haben, wiederholt und in wechselnden Zusammenhängen unter Beweis zu stellen.

Für die Schülerinnen und Schüler sollen ein den Lernprozess begleitendes Feedback sowie Rückmeldungen zu den erreichten Lernständen eine Hilfe für die Selbsteinschätzung sowie eine Ermutigung für das weitere Lernen darstellen.

Die Leistungsrückmeldung erfolgt in regelmäßigen Abständen (zumindest zum Quartalsende) in schriftlicher oder mündlicher Form. Zu umfangreicheren Arbeiten im Bereich der Sonstigen Mitarbeit (z.B. Referate, Projektarbeiten) erfolgt eine zeitnahe Leistungsrückmeldung. Die Beurteilung von Leistungen soll grundsätzlich mit der Diagnose des erreichten Lernstandes und Hinweisen zum individuellen Lernfortschritt verknüpft sein.

Die Kriterien der Notengebung sollen für die Schülerinnen und Schüler transparent sein und ihnen zu Beginn des Schuljahres vorgestellt werden, bei Lehrerwechsel auch zu Halbjahresbeginn. Zur Herstellung von Transparenz in der Leistungsbewertung gehört die Klärung des Kompetenzbegriffs im Fach Informatik und seiner Kompetenzbereiche. Dazu gehört, dass Schülerinnen und Schüler explizit darüber informiert werden, dass alle Kompetenzbereiche zu entwickeln und bei der Leistungsbewertung angemessen berücksichtigt werden. Es wird mit ihnen geklärt, welche Leistungssituationen und -möglichkeiten der Unterricht enthalten wird.

Im Sinne der dem Fach zugrundeliegenden Leitlinien sind alle ausgewiesenen Kompetenzbereiche (Sach-, Methoden-, Urteils- und Handlungskompetenz) bei der Leistungsbewertung angemessen zu berücksichtigen.

Überprüfungsformen schriftlicher, mündlicher und ggf. praktischer Art sollen deshalb darauf ausgerichtet sein, die Erreichung der Kompetenzerwartungen zu überprüfen. Ein isoliertes, lediglich auf Reproduktion angelegtes Abfragen einzelner Daten und Sachverhalten allein kann dabei den zuvor formulierten Ansprüchen an die Leistungsfeststellung nicht gerecht werden.

Bei nicht ausreichenden Leistungen bietet die Lehrkraft dem Schüler bzw. der speziellen Beratungstermine an. Zentrale Inhalte der Beratungsgespräche werden dokumentiert. Zudem werden die Lernhinweise und die Unterstützungsangebote der Lehrkraft schriftlich festgehalten.

Eltern erhalten bei Elternsprechtagen sowie im Rahmen regelmäßiger Sprechstunden Gelegenheit, sich über den Leistungsstand ihrer Kinder zu informieren und dabei Perspektiven für die weitere Lernentwicklung zu besprechen. Die im Fach Informatik angestrebten Kompetenzen umfassen auch Werturteile, Haltungen und Verhaltensweisen, die sich einer unmittelbaren Lernerfolgskontrolle entziehen. Als Informatiklehrerinnen und Informatiklehrer achten wir darauf, dem schulischen Leistungsprinzip das christliche Menschenbild gegenüberzustellen und damit die Relativität von Leistung zu verdeutlichen.

Anzahl und Dauer der Klausuren

Stufe	EF	EF	Q1.1	Q1.2	Q2.1	Q2.2	(Abitur)
Anzahl	1	1	2	2	2	1	1
Dauer/ min	90	90	90	90	135	180	180

3.3.2. Beurteilungsbereich „Klausuren“

Klausuren dienen der schriftlichen Überprüfung der Lernergebnisse in einem Kursabschnitt. Sie sollen darüber Aufschluss geben, inwieweit im laufenden Kursabschnitt gesetzte Ziele erreicht

worden sind. Sie werden nach den Vorgaben der Richtlinien Informatik für die SII korrigiert und beurteilt. Sie bereiten auf die komplexen Anforderungen in der Abiturprüfung vor.

Im zweiten Halbjahr des ersten Jahres der Qualifikationsphase (Q1) kann eine Klausur durch eine Facharbeit ersetzt werden. Die Kriterien zur Bewertung der Facharbeit sind in den Richtlinien Informatik der SII festgelegt.

Die Bewertung der Klausuren erfolgt nach dem folgenden Schema:

Einführungsphase:

Note	sehr gut (1)	gut (2)	befriedigend (3)	ausreichend (4)	mangelhaft (5)	ungenügend (6)
ab ca.	86%	71%	56%	41%	21%	0%

Eine Angabe von Notentendenzen (plus/ minus) ist bei der Benotung der Klausuren möglich.

Qualifikationsphase:

Punkte	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Note	1+	1	1-	2+	2	2-	3+	3	3-	4+	4	4-	5+	5	5-	6
ab ca. [%]	96	91	86	81	76	71	66	61	56	51	46	41	34	27	20	0

Der defizitäre Bereich beginnt bei der Note „ausreichend minus“. Formalfehler und Mängel in der Darstellung werden mit bis zu 15% der Gesamtpunktzahl gewichtet.

3.3.3. Beurteilungsbereich „Sonstige Mitarbeit“

Im Beurteilungsbereich „Sonstige Mitarbeit“ sind alle Leistungen zu werten, die eine Schülerin oder ein Schüler im Zusammenhang mit dem Unterricht mit Ausnahme der Klausuren und der Facharbeit erbringt.

Dazu gehören Beiträge zum Unterrichtsgespräch, beim selbstständigen Arbeiten, in Gruppenarbeit, bei der Mitarbeit in Projekten sowie bei der Präsentation von Arbeitsergebnissen.

Die Schülerinnen und Schüler werden durch die verschiedenen Formen der „Sonstigen Mitarbeit“ auf die mündliche Abiturprüfung vorbereitet und lernen deren Struktur und Beurteilungskriterien kennen.

Die in 1.4.3 für die SI genannten Teilbereiche der sonstigen Mitarbeit gelten auch für die Oberstufe. Nach den Richtlinien Informatik für die SII sind die Beiträge der Schülerinnen und Schüler zum Unterrichtsgespräch Basis der Leistungsbewertung in diesem Beurteilungsbereich. Zudem hat das Vortragen der Hausaufgaben im Fach Informatik einen beachtlichen Stellenwert.

Verstärkt sollen jedoch in der Oberstufe Selbstständiges Arbeiten (auch und insbesondere in Vertretungsstunden) sowie Arbeiten in Gruppen und Projekten gefordert und gefördert werden. Bei der Leistungsbewertung in diesen Arbeitsformen können Gesichtspunkte sein, wie und in welchem Umfang die Schülerinnen und Schüler

- Beiträge zur Arbeit leisten
- Beiträge anderer aufnehmen und weiterentwickeln
- sich in die Denkweisen anderer einfinden
- Aufgaben wie Gesprächsleitung, Protokollführung, Programmorganisation übernehmen
- Informationen beschaffen und erschließen
- ihre Gruppenarbeit organisieren und durchführen, auch in arbeitsteiligen Verfahren
- systematische und heuristische Vorgehensweisen nutzen
- ihre Arbeitsschritte überprüfen, diskutieren und dokumentieren

Bei der selbstständigen Arbeit kann darüber hinaus mit bewertet werden, inwieweit eine Schülerin oder ein Schüler in der Lage ist

- das eigene Lernen zielbewusst zu planen und zu steuern
- den eigenen Lernerfolg zu überprüfen
- daraus Rückschlüsse zu ziehen für das weitere Lernen

4. Entscheidungen zu fach- und unterrichtsübergreifenden Fragen und Beitrag zur UNESCO-Schule

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

Zusammenarbeit mit anderen Fächern

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

Insbesondere stellen die Fächer *Mathematik*, *Physik* und *Sozialwissenschaften* eine Fülle von Problemen bereit, deren Behandlung den Informatikunterricht bereichern. Genannt seien hier:

Mathematik: Primzahlbehandlung, Sieb des Eratosthenes, Gauß-Algorithmus zur Lösung von Gleichungssystemen, Behandlung von Wachstumsprozessen, Grafische Auswertung von statistischem Datenmaterial, Berechnung von Ausgleichsgeraden, Zahlensysteme (Binärsystem, Hexadezimalsystem), Auswertung mathematischer Terme durch Syntaxanalyse, Funktionsplotter zur grafischen Darstellung von Schaubildern, numerische (näherungsweise) Lösung von Gleichungen, Kurvendiskussion

Physik: Simulation physikalischer Vorgänge, z.B. schiefer Wurf mit Luftreibung

Sozialwissenschaften: Probleme des Schutzes personenbezogener Daten, Datenschutzgesetz, Möglichkeiten zur Identitätsermittlung aus wenigen Daten

Fachspezifischer Beitrag zur UNESCO-Schule:

Diskussion zu Freiheiten, Chancen und Problemen im digitalen Zeitalter.

5. Qualitätssicherung und Evaluation

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

Das schulinterne Curriculum (siehe 2.1) ist zunächst bis 2022 verbindlich. Anschließend werden in einer Sitzung der Fachkonferenz Erfahrungen ausgetauscht und ggf. Änderungen für den nächsten Durchgang beschlossen, um erkannten ungünstigen Entscheidungen schnellstmöglich entgegenwirken zu können.

Nach Abschluss des Abiturs 2022 wird die Fachkonferenz Informatik auf der Grundlage ihrer Unterrichtserfahrungen eine Gesamtsicht des schulinternen Curriculums vornehmen und ggf. eine Beschlussvorlage für die erste Fachkonferenz des folgenden Schuljahres erstellen.

6. Anhang

Die Tabelle auf den folgenden Seiten gibt brauchbare Kriterien für die Bildung der Note im Bereich „Sonstige Mitarbeit“.



KRITERIEN FÜR DIE LEISTUNGSBEWERTUNG DER SONSTIGEN MITARBEIT IM INFORMATIK- UNTERRICHT DER SEK. II

<u>Note/ Punkte</u>	Unterrichtsgespräch und Sachkompetenz	Hausaufgaben	Methodenkompetenz	Verhalten bei Gruppenarbeit, Sozialkompetenz	Selbstkompetenz
sehr gut/ 13-15	gleichmäßig hohe, konzentrierte und äußerst qualitätvolle Mitarbeit im Unterricht; Erkennen eines Problems und dessen Einordnung und eigenständige Lösung in einen größeren Zusammenhang; sachgerechte und ausgewogene Beurteilung; angemessene, klare sprachliche Darstellung und richtige Verwendung von Fachbegriffen, Algorithmen und Strukturen	HA und Projekte regelmäßig, differenziert und gründlich mit herausragenden Ergebnissen	methodische Vielfalt; zielsicheres Beschaffen von Informationen und deren Verarbeitung; überzeugende Präsentation auch von Teilergebnissen; besonders strukturierte und inhaltliche Qualität; sachgerechter, souveräner Einsatz fachspezifischer Arbeitstechniken	bringt bei Schwierigkeiten die ganze Gruppe voran; übernimmt Verantwortung für die Gruppe, unterstützt die anderen hervorragend	zeigt eine ausgeprägte Bereitschaft, sich auf Fragestellungen des Unterrichtes einzulassen und sich mit diesen korrelativ sowie kritisch auseinanderzusetzen; anstrengungsbereit, intrinsisch motiviert; selbstbewusst, nicht leicht zu entmutigen
gut/ 10-12	hohe Mitarbeit im Unterricht; Verständnis schwieriger Sachverhalte und deren Einordnung in den Gesamtzusammenhang des Themas; Erkennen des Problems; Kenntnisse über die Unterrichtsreihe hinaus; Verwendung von Fachbegriffen; vermittelte Fachkenntnisse werden beherrscht; unterrichtsfördernde Beiträge; weitgehend vollständige weiterführende Ausführungen; sicherer Umgang auch mit anspruchsvollen Problemen und sachgerechte Anwendung von Fachtermini;	HA und Projekte regelmäßig und differenziert mit guten Leistungen	selbstständige Informationsbeschaffung; Kenntnis und souveräne Anwendung verschiedener Arbeits- und Präsentationstechniken; sachgerechter, souveräner Einsatz fachspezifischer Arbeitstechniken	aktiv an zeitökonomischer Planung und Durchführung der GA beteiligt, wirkt bei Schwierigkeiten aktiv an Problemlösung mit, geht aktiv auf Meinungen anderer ein	zeigt eine ausgeprägte Bereitschaft, sich auf Fragestellungen des Unterrichtes einzulassen und sich mit diesen korrelativ auseinanderzusetzen; traut sich auch schwierige Aufgaben zu; Gedankengänge werden selbstständig weiterentwickelt und klar dargestellt und argumentativ vertreten



<p>befriedigend/ 7-9</p>	<p>insgesamt regelmäßig freiwillige Mitarbeit im Unterricht; im Wesentlichen richtige Wiedergabe einfacher Fakten und Zusammenhänge aus unmittelbar behandeltem Stoff; vermittelte Fachkenntnisse werden überwiegend beherrscht; weitgehend vollständige, differenzierte, sachbezogene Ausführungen; mittelschwere Algorithmen werden sachgerecht verstanden und bearbeitet; Darstellungen in der Programmiersprache sind weitgehend brauchbar; begrenztes Problembewusstsein</p>	<p>HA und Projekte regelmäßig und mit befriedigenden Leistungen</p>	<p>kann projektdienliche Informationen einbringen, zugeteilte Inhalte erfassen und dokumentieren; fachspezifische Arbeitstechniken werden meist sachgerecht ausgewählt und eingesetzt; Verwendung adäquater Arbeitsmaterialien</p>	<p>erkennbare Mitverantwortung für das gemeinsame Projekt; sorgt mit für störungsfreies Miteinander; zeigt Bereitschaft, eigene und Gruppenergebnisse zu präsentieren</p>	<p>zeigt Bereitschaft, sich auf Fragestellungen des Unterrichtes einzulassen und sich mit diesen korrelativ auseinanderzusetzen; lässt sich nicht leicht entmutigen; greift gelegentlich Beiträge anderer auf und führt diese fort; verständliche Ausdrucksweise und leicht nachvollziehbare Programmteile mit hinreichenden Kommentaren</p>
<p>ausreichend/ 4-6</p>	<p>nur gelegentlich freiwillige Mitarbeit im Unterricht; Äußerungen beschränken sich auf die Wiedergabe einfacher Fakten und Zusammenhänge aus dem unmittelbar behandelten Stoffgebiet und sind im Wesentlichen richtig; vermittelte Fachkenntnisse werden mit Einschränkungen beherrscht; leichtere Programmstücke werden dem Sinn nach richtig erfasst; teilweise nachlässige Darstellungen in der Programmiersprache</p>	<p>HA und Projekte nicht regelmäßig oder nur oberflächlich erledigt</p>	<p>Schwierigkeiten, Arbeitsabläufe folgerichtig und zeitökonomisch zu planen, fachspezifische Arbeitstechniken (z.B. Programmierung; Umsetzung v. Algorithmen; Programmanalyse) werden eingesetzt</p>	<p>Kommunikationsfähigkeit und -bereitschaft nur in Ansätzen; verlässt sich lieber auf die anderen Gruppenmitglieder; beteiligt sich zuweilen aktiv an Entscheidungsprozessen in der Gruppe; selten Präsentation von Ergebnissen</p>	<p>zeigt Bereitschaft, sich auf Fragestellungen des Unterrichtes einzulassen; neigt bei auftretenden Schwierigkeiten zu ausweichendem Verhalten, ist auf Nachfrage in der Lage, sich zu den Beiträgen anderer zu äußern; folgt den Beiträgen anderer noch aufmerksam</p>



<p>mangelhaft/ 1-3</p>	<p>überwiegend passives Verhalten im Unterricht; Äußerungen nach Aufforderung sind einsilbig, unstrukturiert und nur teilweise richtig; Fehlleistungen, auch nach Vorbereitung; Schwierigkeiten bei sachbezogener Verarbeitung von Wissen und der Verknüpfung von Zusammenhängen; oft mangelndes Verständnis der eigentlichen Probleme</p>	<p>HA oder Projekte häufig nicht vorhanden oder nur lückenhaft erledigt; zu vorbereiteten Themen meist keine sachbezogenen Äußerungen</p>	<p>kaum in der Lage, mit den Lerngegenständen sachgerecht und systematisch umzugehen; legt lediglich unverarbeitetes Material (z.B. Internetausdrucke oder abgeschrieben Programmteile) vor</p>	<p>wenig projektdienliche Mitarbeit; wenig zuverlässig; nicht auf Gruppenarbeit vorbereitet; übernimmt keine Mitverantwortung; beschränkt Rolle am liebsten auf „Programmtester“ für die Gruppe</p>	<p>kaum Bereitschaft, sich auf Fragestellungen des Unterrichtes einzulassen</p>
<p>ungenügend/ 0</p>	<p>keine freiwillige Mitarbeit im Unterricht; Äußerungen nach Aufforderung sind falsch; Fachwissen nicht erkennbar</p>	<p>sehr häufig keine HA, fehlende Projektarbeit</p>	<p>keine Arbeitsplanung; nicht in der Lage, mit den Lerngegenständen sachgerecht umzugehen; nicht in der Lage, Informationen einzuholen und darzustellen</p>	<p>kein situationsangemessenes Gesprächsverhalten; nicht kooperativ; bei Gruppenarbeit kein Interesse an eigenem Arbeitsanteil und an Arbeitskontakten zu Mitschülern; hält sich nicht an Regeln</p>	<p>keine Bereitschaft, sich auf Fragestellungen des Unterrichtes einzulassen; fehlende Selbständigkeit im Arbeiten; Abschreiben der Arbeitsergebnisse der Mitschüler; wiederholtes unentschuldigtes Fehlen</p>